

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO ADMINISTRACI KOMERČNÍ SÍŤE

BAKALÁŘSKÁ PRÁCE

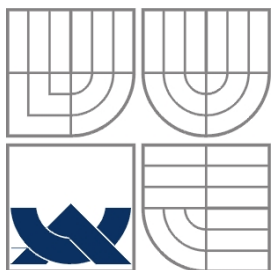
BACHELOR'S THESIS

AUTOR PRÁCE

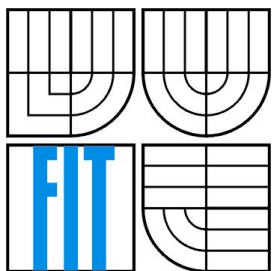
AUTHOR

MICHAL PÁPAI

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO ADMINISTRACI KOMERČNÍ SÍTĚ

SYSTEM FOR ADMINISTRATION OF COMMERCIAL COMPUTER NETWORK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL PÁPAI

VEDOUCÍ PRÁCE
SUPERVISOR

MGR. ROMAN TRCHALÍK

BRNO 2009

Abstrakt

Tato bakalářská práce se zabývá vývojem komplexního systému pro administraci komerční sítě. Tvorba uvedeného systému je založena na zkušenostech z provozu skutečné počítačové sítě poskytovatele internetového připojení, který využívá jako hlavní bránu server s OS Linux. Práce odhaluje nedostatky současného řešení. Na jejich základě se snaží navrhnout a implementovat jednotné GUI pro administraci této sítě, jednotlivé části automatizovat, snížit odbornou znalost obsluhy systému a provádět automatickou kontrolu.

Klíčová slova

ISP, fakturační systém, linux, směrovač, řízení šířky pásma, PHP, MySQL

Abstract

This bachelor's thesis describes the development of the system for administration of commercial computer network. The development is based on real experiences of the Internet service provider and its computer network with Linux-based gateway. This thesis deals with shortcomings of the existing solution and tries to propose and implement a uniform GUI for administration, to automatize individual parts and action, to reduce the professional knowledge of the system and perform automatic control of this kind of network.

Keywords

ISP, billing system, Linux, router, gateway, traffic control, PHP, MySQL

Citace

Michal Pápai: Systém pro administraci komerční sítě, bakalářská práce, Brno, FIT VUT v Brně, 2009

Systém pro administraci komerční sítě

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Mgr. Romana Trchalíka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Pápai
20.05.2009

Poděkování

Rád bych tímto poděkoval Mgr. Romanu Trchalíkovi za poskytnuté rady a odborné vedení celé práce.

© Michal Pápai, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Hlavný úvod.....	2
1.1 Stručný prehľad.....	2
2 Súčasný stav siete.....	3
2.1 Predstavenie.....	3
2.2 OS Linux ako gateway.....	3
2.2.1 Firewall a preklad adres.....	3
2.2.2 Riadenie šírky pásma.....	7
2.3 Nedostatky súčasného riešenia.....	9
3 Návrh systému.....	10
3.1 Analýza požiadaviek.....	10
3.1.1 Užívatelia a ľudský faktor.....	10
3.1.2 Funkcionalita.....	11
3.1.3 Rozhranie systému.....	19
3.1.4 Zdroje.....	19
3.1.5 Dokumentácia.....	20
3.2 Use case model.....	20
3.2.1 Aktéri.....	20
3.2.2 Prípady použitia.....	21
4 Implementácia systému.....	26
4.1 Realizácia kľúčových častí systému.....	26
4.1.1 Registrácia a aktivácia prípojky.....	27
4.1.2 Blokovanie prístupu.....	30
4.1.3 Účtovanie poplatkov.....	33
4.1.4 Štatistiky a grafy.....	34
4.2 Implementácia a použité technológie.....	35
4.2.1 Programovací jazyk.....	35
4.2.2 Databázový systém.....	36
4.2.3 Štruktúra projektu.....	36
5 Záver.....	37
5.1 Možnosti ďalšieho rozšírenia.....	37
Literatúra.....	38
Zoznam príloh.....	39

1 Hlavný úvod

Nebolo to tak dávno, keď bol internet aj v našej krajine neprebádanou novinkou z pohľadu domácich užívateľov a každý „vlastník“ internetu bol považovaný za nadpriemerne bohatého, ktorý pre svoj tuzemský život potrebuje kadejaký rozmar. V dobe vytáčaného (dial-up) internetového pripojenia to bolo ťažké hlavne, keď si to pozrieme z dnešného pomeru cena/výkon. Závratná rýchlosť možno niekedy dosahujúca 5kB/s neumožňovala toho viac ako poslať textové e-maily alebo prezerať jednoduché web stránky. Avšak i v tejto sfére ako každej inej napreduje vývoj kilometrovými krokmi. S pribúdaním nových technológií rástol aj počet užívateľov internetového pripojenia a tak sa dostal takmer do každej domácnosti a kútov sveta. Zatiaľ posledným míľnikom v tomto smere sa stalo poskytovanie širokopásmového internetu či už metalickou alebo bezdrôtovou technológiou. Takéto pripojenia nemusia poskytovať gigantický obri ako sú veľké telekomunikačné firmy, ale i ľudia s chuťou do práce, menšie a stredné firmy tzv. lokálni poskytovatelia internetu. V blízkej budúcnosti, ako niektorí odborníci predpokladajú, takáto expanzia siete internet a digitalizácia nahradí klasickú televíziu v podobe Triple-Play¹ služieb. Narastanie počtu pripojených klientov nesie so sebou aj radu problémov, hlavne pre poskytovateľa pripojenia (ISP) respektíve pre jeho zamestnancov, ktorí potrebujú rýchly a efektívny systém správy siete so širokou škálou informácií a funkcií. Ako to väčšinou býva, každá menšia a stredne veľká firma sa snaží nasadiť voľne dostupné nástroje a techniky pre vytvorenie a správu siete za účelom minimalizovať náklady, a v neposlednom rade nasadiť rôzne druhy informačných systémov na evidenciu zákazníkov.

Táto práca sa zaoberá automatizáciou niektorých úkonov na strane ISP - ako je evidencia klientov v systéme, platobný systém, centrálna konfigurácia siete, atď. s cieľom vytvorenia komplexného IS. Od prípadného čitateľa sa očakáva základná znalosť sieťových technológií a orientácia v OS Linux.

1.1 Stručný prehľad

- Kapitola jedna je úvod a práve ju čítate.
- Druhá kapitola predstavuje lokálne pôsobiaceho ISP a popisuje použité nástroje pre vytvorenie hlavnej brány ISP a problematiku zabezpečenia šírky pásma. Taktiež je tu vytýčený cieľ práce s ohľadom na nedostatky súčasného riešenia.
- Ďalšia kapitola sa zaoberá detailnejšou analýzou požiadaviek na IS zo strany ISP, predovšetkým jeho funkcionalitou.
- Štvrtá kapitola poukazuje na možnosť riešenia, kde je naznačená realizácia kľúčových častí požadovaného systému, a použité programovacie prostriedky z hľadiska implementácie.
- Posledná kapitola je záver a obsahuje zhodnotenie vyvíjaného systému s možnosťami jeho ďalšieho rozšírenia.

1 http://sk.wikipedia.org/wiki/Triple_Play

2 Súčasný stav siete

2.1 Predstavenie

SURANY.net²s.r.o. je lokálnym poskytovateľom internetového pripojenia pôsobiaci v meste Šurany a v okolitých dedinách. ISP poskytuje WiFi pripojenie podľa štandardov IEEE 802.11a/b/g a taktiež pripojenie pomocou metalického vedenia podľa štandardu IEEE 802.3u. V čase písania tejto práce je súčasný celkový počet pripojených klientov takmer 1000 a konektivita do siete internet 150Mbit/s.

Ako hraničný smerovač je použitý OS Linux, ktorý beží na serveri s nasledujúcou konfiguráciou:

- Dual Core Intel® Xeon® X3065, 2.33GHz, 2x2MB Cache, 1333MHz
- 2GB DDR2 SDRAM 667MHz Memory (2x1GB Dual Ranked DIMMs), ECC
- 2x 250GB 7.200 rpm 3.5-inch SATA Hard Drive
- Dual Broadcom® Gigabit Ethernet NIC

Táto práca sa ďalej nezaobrá a nepopisuje bližšie topológiu siete, ale snaží sa na základe súčasného stavu navrhnuť a implementovať SW, ktorý by umožnil evidenciu klientov a s tým spojenú správu takéhoto smerovača do jednotného komplexného systému.

2.2 OS Linux ako gateway

Linux je voľne šíriteľný operačný systém s prístupnými zdrojovými kódmi. Jeho pôvodná verzia bola v priebehu desiatich rokov zdokonaľovaná nespočetným množstvom ľudí na celom svete. Práve dostupnosť zdrojových kódov robí z tohto systému dokonalý operačný systém a umožní rýchle odstránenie prípadných chýb zo systému. Opravy a vylepšenia sú k dispozícii prostredníctvom Internetu v priebehu niekoľkých dní. Licenčné podmienky OS Linuxu sú pomerne jednoduché a prijateľné pre každého, vychádzajú z licencie GPL (General Public Licence) [1].

OS Linux vďaka veľkej podpore sieťových služieb je najčastejšie využívaný ako server respektíve v našom prípade ako brána (gateway) do siete internet.

2.2.1 Firewall a preklad adries

Základným prvkom bezpečnosti v každej sieti je firewall. V tomto projekte je použitý program iptables [7], ktorý je implicitným firewallom takmer vo všetkých distribúciách OS Linux. Iptables je nástupníkom ipchain od rovnakej organizácii Netfilter a obsahuje sadu vylepšených vlastností ako sú:

- lepšia integrácia s linuxovým jadrom čo vedie vyššej spoľahlivosti a rýchlosti
- stavový paketový filter ktorý umožňuje sledovať každé jedno spojenie a invokovať tak akcie potrebné pre niektoré typy sieťových služieb (napr. aktívne FTP)
- filtrovanie paketov založené na kontrole MAC adresy a hlavičiek paketov
- lepšia podpora pre všetky typy prekladu sieťových adries (NAT/NAPT pre IPv4)

2 <http://www.surany.net>

- logovanie ktoré poskytuje možnosť prispôbiť úroveň podrobnosti o podávaní správ
- funkcie pre obmedzenie niektorých typov útoku zamietnutie služby (DoS)
- API a veľký počet rozširujúcich modulov

Nakoľko predstavená sieť nemá dostatok verejných IPv4 adries pre pridelenie klientom, je nutné zaviesť preklad adries. Preklad adries (NAT³ – Network Address Translation) je veľmi široký pojem pod ktorým je zoskupených niekoľko typov prekladu adries. Vo všeobecnosti NAT definuje statické alebo dynamické prekladanie vnútorných adries siete na vonkajšie adresy ktoré majú ďalej prístup do inej časti siete alebo internetu.

Pre príklad: na vnútornom rozhraní je sieť 192.168.1.0/24 a na vonkajšom 87.244.192.0/24; ak paket smeruje z vnútornej siete do vonkajšej (klient požaduje zobrazenie web stránky na internete s adresou 147.229.2.9), zmení sa zdrojová IP adresa v hlavičke paketu z 192.168.1.X na 87.244.192.Y. Web server spracuje daný paket a odpoveď zašle na zdrojovú adresu 87.244.192.Y. Paket prijatý smerovačom, na ktorom bol prevedený preklad vyhladá v NAT tabuľke k akej vnútornej adrese sa viaže vonkajšia IP adresa. Nikto okrem smerovača nevie, kto je v skutočnosti odosielateľom dát, čo je istá forma zabezpečenia identity klienta. Tento typ NATu je možné použiť pokiaľ je dostatočný počet verejných IP adries prekladu 1:1 k privátnym adresám.

Podobným princípom pracuje i typ PAT (Port Address Translation) ktorý pri preklade využíva i porty. Tento typ je využitý i v tomto projekte, pretože na jednu verejnú IP adresu je možné preložiť viacero privátnych. Viac informácií je možné nájsť v dokumentácii firewallu [7].

2.2.1.1 Proces spracovania paketu firewallom iptables

Všetky pakety vstupujúce a vystupujúce z rozhrania smerovača prechádzajú sledom vstavaných tabuliek (front). Každá z týchto front je venovaná určitému typu činnosti, ktorá obsahuje filtrovaciu a transformačnú reťazec (ang. „chains“). Iptables má vstavané 3 typy tabuliek:

- **filter** – obsahuje filtrovaciu pravidlá
- **nat** – tabuľka zodpovedná za preklad sieťových adries
- **mangle** – obsahuje pravidlá pre úpravu paketov a je zodpovedná za zmenu kvality služieb

Pakety postupne prechádzajú všetkými reťazcami v uvedených tabuľkách, až kým jedno z nich nerozhodne o ďalšej akcii. Cesta paketom je taktiež prehľadne zobrazená v diagrame ktorý je súčasťou prílohy č. 1. Jednotlivé reťazce a ich funkcie sú popísané v tabuľke č. 1

3 http://en.wikipedia.org/wiki/Network_address_translation

Typ tabuľky	Funkcia tabuľky	Názov reťazca	Popis
filter	Filtrovanie paketov	INPUT	Reťaz pravidiel pre pakety prichádzajúce pre firewall.
		OUTPUT	Reťaz pravidiel pre pakety, ktoré vznikli a odchádzajú z firewallu
		FORWARD	Reťaz pravidiel pre pakety, ktoré firewallom iba prechádzajú a sú smerované ďalej
nat	Preklad sieťových adries	PREROUTING	Reťaz pravidiel, ktoré sa aplikujú na pakety pri vstupe do firewallu ešte pred samotným smerovaním. Možnosť aplikovania NAT pre zmenu cieľovej adresy. Taktiež známe ako DNAT
		POSTROUTING	Reťaz pravidiel, ktoré sa aplikujú na pakety pri opúšťaní firewallu. Možnosť aplikovania NAT pre zmenu zdrojovej adresy použitím niektorého typu prekladu. Známe ako SNAT
		OUTPUT	Možnosť aplikovania NAT pre pakety generované firewallom.
mangle	Modifikácia TCP paketov	všetky typy	Modifikácia hlavičiek TCP paketov pre zaistenie QoS predtým než dôjde k smerovaniu paketu.

Tabuľka 1: Spracovanie paketov jednotlivými reťazcami

2.2.1.2 Špecifikácia akcie

Každé pravidlo zadaneé vo firewally kontroluje IP pakety a snaží sa tak urobiť ďalšiu akciu, ktoré vyhovie stanoveným podmienkam. Pomocou parametra "-j akcia" je možné určiť osud paketu. Jednotlivý prehľad akcií je prehľadne popísaný v tabuľke č.2

Akcia	Popis	Rozširujúce voľby
ACCEPT	Paket je akceptovaný	-
DROP	Paket je zahodený bez informácie odosielateľa	-
LOG	Informácia o pakete je poslaná na syslog server	--log-prefix "string"
REJECT	Paket je zahodený, odosielateľovi je zaslaný ICMP paket o chybe	[--reject-with icmp- port- unreachable]
DNAT	modifikuje cieľovú adresu alebo port paketu	--to-destination <ipadresa>[:<port>]
SNAT	modifikuje zdrojovú adresu alebo port paketu	--to-source <adresa>[-<adresa>] [:<port>-<port>]
MASQUERADE	zmení zdrojovú adresu odchádzajúceho paketu na adresu sieťového rozhrania, ktorým opustil firewall	[--to-ports <port>[-<port>]]

Tabuľka 2: Popis najčastejších akcií pri zhode pravidla

2.2.1.3 Dôležité parametre a kritéria pre pravidlá

Jednotlivé pravidlá pre firewall iptables sa vytvárajú resp. odstraňujú volaním príkazu `iptables` z príkazovej riadky. Nasledujúce dve tabuľky č. 3 a č. 4 zobrazujú prehľad najčastejšie používaných prepínačov pre prácu s firewallom.

Prepínač	Popis
-t <tabuľka>	Tabuľka nad ktorou sa aplikuje pravidlo. Implicitná je tabuľka <i>filter</i>
-j <akcia>	Akcia ktorá bude prevedená po splnení podmienky pravidla
-A <reťazec>	Pridá pravidlo na koniec reťazca
-F	Vymaže všetky pravidlá nad zvolenou tabuľkou
-p <typ-protokolu>	Pravidlo pre protokol. Typ môže byť "tcp", "udp", "icmp" alebo "all"
-s <ip-adresa>	Pravidlo pre zdrojovú adresu. Môže byť konkrétna IP adresa alebo sieť v tvare adresa/maska
-d <ip-adresa>	Pravidlo pre cieľovú adresu. Môže byť konkrétna IP adresa alebo sieť v tvare adresa/maska
-i <nazov-rozhrania>	Pravidlo pre podmienku vstupného rozhrania
-o <nazov-rozhrania>	Pravidlo pre podmienku výstupného rozhrania

Tabuľka 3: Prepínače pre pravidlá firewallu

Prepínač	Popis
-p tcp/udp --sport <port>	Podmienka pre zdrojový port. Môže byť hodnota alebo rozsah od:do
-p tcp/udp --dport <port>	Podmienka pre cieľový port. Môže byť hodnota alebo rozsah od:do
-p tcp --syn	Používané pre identifikáciu vytvorenia nového TCP spojenia

Tabuľka 4: Prepínače pre špecifikáciu kritérií pre TCP/UDP protokol

2.2.1.4 Príklady

Predchádzajúce kapitoly stručne ukázali na možnosť práce s paketovým firewallom iptables. Ako sa pravidlá zapisujú a aký majú význam bude viac zrejmé z nasledujúcich príkladov:

- **Povolenie IP adresy 192.168.1.1 pre smerovanie**

```
iptables -A FORWARD -i eth0 -s 192.168.1.1 -o eth1 -d 0/0 -j ACCEPT
iptables -A FORWARD -i eth1 -s 0/0 -o eth0 -d 192.168.1.1 -j ACCEPT
```

- **Vytvorenie PAT pre vnútornú sieť 192.168.1.0/24 na verejnú IP 87.244.2.200**

```
iptables -t nat -A POSTROUTING -s 192.169.1.0/24 -j SNAT --to 87.244.2.200
```

- **Vytvorenie NAT pre IP 10.10.10.1 na verejnú IP 87.244.2.1**

```
iptables -t nat -A POSTROUTING -s 10.10.10.1 -j SNAT --to 87.244.2.1
iptables -t nat -A PREROUTING -d 87.244.2.1 -j DNAT --to 10.10.10.1
```

2.2.2 Riadenie šírky pásma

Ani jedna sieť nemá neobmedzené prostriedky čo sa týka priepustnosti. Ak chceme zabezpečiť zákazníkom určitú šírku pásma zo svojej konektivity, musíme definovať pre zákazníka minimálnu (garantovanú) priepustnosť, ktorú zákazník má v každom prípade dostupnú, a maximálnu na požadovanú hodnotu šírku pásma. Ak by bol o šírku pásma väčší záujem než sme schopný poskytnúť prichádza k vzájomnému obmedzovaniu. Takéto obmedzovanie vieme v OS Linux kontrolovať pomocou radiaceho mechanizmu paketov qdisc (queueing discipline). Samotné nastavovanie šírky pásma je umožnené vďaka implementácii TCP/IP a akým spôsobom pracuje. Predstavme si server a klienta ktorý vzájomne komunikujú. Po ustavení spojenia zvyčajne server posiela dáta rýchlejšie než sme schopný prijať, a tak sú dáta po ceste ku klientovi zahodené a nedoručené. Server nedostáva potvrdenie o prijatých paketoch a postupne znižuje svoju rýchlosť odosielania. Obdobným spôsobom je možné umelo vyvolať postupné zníženie prenosovej rýchlosti, a tým nastaviť požadovanú šírku pásma, pomocou rôznych radiacích disciplín obsahujúce triedy. V tomto projekte je použitá disciplína Hierarchical Token Bucket (HTB)[8] ktorá je aj východiskom pre nasadený program Prometheus QoS.

2.2.2.1 Programy pre zabezpečenie šírky pásma

Zadávanie pravidiel pre jadro OS je možné pomocou programu tc (Traffic Control) z balíka iproute. Vytváranie jednotlivých tried je hierarchické, a ako prvý bod je nutné vytvoriť koreňový qdisc. Tento uzol slúži ako kontajner pre celkový traffic a taktiež definuje ktorý radiaci algoritmus bude použitý. Ďalej sa vytvorí trieda ktorá bude mať šírku pásma rovnakú ako je konektivita do siete internet. Nad touto triedou môžeme vystavať požadovanú hierarchiu tried, ktoré už konkrétne budú reprezentovať šírku pásma klientov. Základný sled príkazov je na nasledujúcom príklade:

```
tc qdisc add dev eth0 root handle 1:0 htb
tc class add dev eth0 parent 1:0 classid 1:1 htb rate 150000kbit
tc class add dev eth0 parent 1:1 classid 1:10 htb rate 128kbit ceil /
1024kbit prio 0
tc class add dev eth0 parent 1:1 classid 1:20 htb rate 128kbit ceil /
2048kbit prio 0
```

Popis parametrov ovplyvňujúce šírku pásma:

- **rate** – určuje minimálnu šírku pásma ktorú má trieda a jej potomkovia garantovanú
- **ceil** – nastaví pre triedu maximálnu šírku pásma, ktorú je možné dosiahnuť ak je požadovaná kapacita dostupná na nadradenej triede
- **prio** – definuje pre triedu prioritu obsluhy. Čím menšie je číslo tým je priorita väčšia

Na to aby mohli byť pakety zaradené do týchto vytvorených tried, je ich nutné poznačiť firewallom iptables s akciou CLASSIFY:

```
iptables -A POSTROUTING -d 192.168.1.1 -o eth0 -j CLASSIFY --set-class 1:10
```

Horeuvedený príkaz nastaví šírku pásma smerom ku klientovi s IP adresou 192.168.1.1. Pakety smerujúce na túto IP budú zaradené do triedy ktorej maximálna priepustnosť je 1024kbit za sekundu.

2.2.2.2 Prometheus QoS

Prometheus⁴ QoS je program dostupný pod GPL licenciou, ktorý je orientovaný pre ISP. Je to efektívny nástroj na manipuláciu so šírkou pásma pre IP tok a nastavovanie FUP⁵ na základe prenesených dát. Pracuje na rovnakom princípe vytvárania tried ako je ukázané v predchádzajúcej kapitole. Po vytvorení základných tried, podtried a pravidiel pre klasifikáciu paketov pre cca. 1000 klientov smerovač v uvedenej konfigurácii nebol schopný zvládnuť tok paketov a riadenie šírky pásma bolo neúčinné. Tento problém rieši spomínaný program zavedením indexovania. Prometheus vytvára v iptables zložité indexy pre subnety rôznej veľkosti, aby jeden paket neprechádzal všetkými pravidlami (namiesto 2000, iba typicky napríklad 7.) Pre N IP adries je miesto zložitosti $O(N)$ pri klasickom lineárnom prechádzaní pravidiel, zložitost' $\log_2(N)$ podľa autora programu.

Nevýhodou tohto programu je chýbajúca dokumentácia. Na stránkach programu je ale možné stiahnuť príklad konfigurácie s vysvetlením, čo je celkom postačujúce. Implicitný konfiguračný súbor je obsiahnutý v prílohe č. 7. Ďalším záporným rysom je že nie je schopný dynamicky zareagovať napr. pri aplikovaní FUP v počítaní prenesených dát a musí byť v požadovaných intervaloch reštartovaný.

Kľúčové slová (keyword)

Ako je možné vidieť z priloženej konfigurácie program definuje kľúčové slová. Podľa týchto definícií sa odvíja generovanie materských tried, ktoré slúžia pre nastavenie šírky pásma skupine podtried pre klientov. Na každé kľúčové slovo sa viaže rada parametrov ovplyvňujúca šírku pásma. Tabuľka č. 5 popisuje najčastejšie nastavované parametre.

FUP - Fair User Policy

Pre aplikovanie FUP používa Prometheus QoS kreditový systém. Každý klient má pridelený kredit (objem prenesených dát v MB), ktorý keď presiahne za čas medzi prvým a druhým spustením programu, mu bude upravená rýchlosť na jeho minimálnu (rate). Tento kredit môže byť fixne stanovený alebo relatívne vypočítaný z minimálnej rýchlosti pre triedu klienta. Výpočet kreditu je znázornený v tabuľke č. 5.

parameter	Popis
asymetry-ratio	Konštanta pre pomer nastavenia šírky pásma pre upload. Upload je vypočítaný podľa <i>download/asymetry-ratio</i>
asymetry-treshold	Hodnota v kbps udávajúca o koľko bude upload menší než download
magic-relative-limit	Konštanta pre nastavenie kreditu v závislosti od garantovanej šírky pásma. Po prekročení kreditu bude znížená šírka pásma ceil na hodnotu rate Kredit je vypočítaný podľa <i>magic-relative-limit*min_šírka_pásma</i> (v MB)
magic-relative-prio	Konštanta pre nastavenie kreditu v závislosti od garantovanej šírky pásma. Po prekročení kreditu bude znížená priorita obsluhy paketov.
magic-fixed-limit	Pevne stanovený kredit pre šírku pásma.
magic-fixed-prio	Pevne stanovený kredit pre prioritu.
htb-default-prio	Implicitná priorita obsluhy pre triedy

Tabuľka 5: Najbežnejšie nastavované parametre pre Prometheus QoS

4 <https://dev.arachne.cz/svn/prometheus/wiki>

5 http://cs.wikipedia.org/wiki/Fair_User_Policy

Program po spustení načíta štandardný linuxový súbor hosts (/etc/hosts). V tomto súbore program očakáva zoznam klientov s definovanými príslušnými kľúčovými slovami a k nim priradenú šírku pásma. Tieto kľúčové slová sú v súbore formou komentára. Syntax zápisu je nasledovná a ukážka je zobrazená na obrázku č.1:

```
IPADRESA    HOSTNAME    #keyword-minrate-maxrate
```

10.102.0.6	10.102.0.6	#backbone-plus-128-90000
10.101.0.8	10.101.0.8	#backbone-plus-128-90000
10.10.51.5	10.10.51.5	#wifi-plus-128-1024
10.10.17.35	10.10.17.35	#wifi-plus-128-2048

Obrázok 1: Ukážka súboru hosts - zápis pre QoS

2.3 Nedostatky súčasného riešenia

V tejto podkapitole je zhrnutý zoznam nedostatkov súčasného riešenia a vyvodený záver pre návrh komplexného systému.

- časová náročnosť obsluhy na prihlásenie sa do systému
- nutná externá evidencia zoznamu klientov kvôli neprehľadnosti súboru hosts
- program Prometheus QoS neumožňuje účtovanie poplatkov
- neprehľad v ponúkaných paušáloch a ich cena
- pri pridávaní riadku (záznamu) do súboru hosts sa ľahko môže stať preklep, ktorý nemusí byť hneď badateľný
- štatistiky generované programom Prometheus sú neprehľadné a nemajú grafickú podobu

Na základe týchto nedostatkov, bude vyvíjaný systém, ktorý má vytvoriť jednotné GUI pre administráciu tejto siete, jednotlivé časti automatizovať, znížiť odbornú znalosť obsluhy systému a prevádzať automatickú kontrolu.

3 Návrh systému

Nasledujúca kapitola obsahuje popis analýzy a návrh systému pre správu siete založenej na OS Linux v predstavenej 2. kapitole. Samotný proces prípravy implementácie projektu je založený na klasickom prístupe obsahujúcom prvé body zo životného cyklu obecného informačného systému definované v [2]

3.1 Analýza požiadaviek

Nasledujúce body analýzy požiadaviek boli získané komunikáciou so zadávateľom projektu predstaveného úvodom 2. kapitoly. Ako prvotná požiadavka bola formulovaná na vytvorenie informačného systému (IS) pre evidenciu a účtovanie poplatkov zákazníkov. Na základe analýzy stávajúceho systému ISP, ktorý plní funkciu smerovača a zároveň je jeho hlavnou bránou do siete internet bola navrhnutá integrácia tohto systému do IS.

V priebehu niekoľkých schôdzok so zadávateľom projektu bol postupne vypracovaný zoznam detailnejších požiadaviek, ktoré boli získané využitím modernej metódy tzv. Brainstorming⁶. Jednotlivé požiadavky vyplývajúce z komunikácie sú rozdelené do logických celkov. Každá požiadavka je charakterizovaná niekoľkými bodmi:

- poradové číslo požiadavky v rámci analýzy
- znenie požiadavky
- priorita pri návrhu a implementácii systému (1- najnižšia... 5- najvyššia)
- predpokladaná náročnosť pri riešení (analýza rizík) (1- najnižšia... 5- najvyššia)

3.1.1 Užívatelia a ľudský faktor

3.1.1.1 Požiadavka č. 1

Systém musí podporovať viac užívateľský prístup s pevne prednastavenými právami pre vstup do jednotlivých modulov systému. Predpokladajú sa 3 úrovne prístupu SuperAdmin, Admin a Technik. Každý užívateľ bude mať v systéme vedené okrem definícii prístupu a prihlasovacích údajov i základné osobné údaje, podľa ktorých ho bude možné jednoznačne identifikovať. Všetky operácie ktoré vedú k zmene stavu systému budú zaznamenávané.

Priorita: 3

Náročnosť: 3

3.1.1.2 Požiadavka č. 2

Užívatelia systému musia mať možnosť zmeny svojich údajov, a nastavenia prístupového hesla.

Priorita: 3

Náročnosť: 3

⁶ <http://en.wikipedia.org/wiki/Brainstorming>

3.1.1.3 Požiadavka č. 3

U všetkých užívateľoch sa predpokladá dostatočná oboznámenosť s problematikou riešenou v rámci projektu, a preto nie je nutné ich rozdelenie s ohľadom na znalosti ktoré majú. (napr. Formou rôznych užívateľských rozhraní). Napriek tomu je požadované vytvorenie krátkej užívateľskej príručky s cieľom oboznámiť sa s používaním systému a uľahčiť im tak začiatky práce s ním.

Priorita: 2

Náročnosť: 3

3.1.2 Funkcionalita

3.1.2.1 Požiadavka č. 4

Užívateľ bude môcť prezerat' ostatných užívateľov systému. Užívateľ s dostatočnými oprávneniami bude ďalej môcť týchto užívateľov editovať (napr. Zmena prístupových práv) alebo odstrániť zo systému, pričom záznamy o užívateľovi v databáze zostanú kvôli integrite, len sa znemožní jeho prihlásenie.

Priorita: 2

Náročnosť: 2

3.1.2.2 Požiadavka č. 5

Pri výpise užívateľov bude systém zobrazovať nasledujúce atribúty: login, meno, priezvisko, dátum, čas a ip adresu posledného úspešného prihlásenia a užívateľova práva. Ďalej okrem zobrazovania úspešného prihlásenia musia byť logované pokusy o prihlásenie.

Priorita: 2

Náročnosť: 2

3.1.2.3 Požiadavka č. 6

Užívateľ bude môcť pridávať a meniť shapovacie profily pre QoS s definovaním nasledujúcich atribútov: meno profilu, asymetrie ratio voči downloadu, limity pre FUP relatívne ku garantovanej rýchlosti. Pri výpise týchto profilov budú tieto parametre zobrazené.

Priorita: 4

Náročnosť: 3

3.1.2.4 Požiadavka č. 7

Systém bude nastavovať šírku pásma do siete internet a účtovať poplatky za služby na základe jednotlivých taríf. Pre tarifu za internet užívateľ bude musieť definovať názov tarify, garantovanú a maximálnu rýchlosť k smerom zákazníkovi, mesačný poplatok a shapovací profil. Podľa uvedených parametrov bude pri výpise zobrazovaný názov tarify, mesačný poplatok, maximálna rýchlosť download a upload, informáciu o tom či je pre danú tarifu aplikovaná FUP, a počet zákazníkov s touto tarifou.

Priorita: 4

Náročnosť: 3

3.1.2.5 Požiadavka č. 8

Okrem internetových taríf systém bude musieť umožniť pridávať i ostatné položky (napr. verejná IP adresa). Pre tieto tarify je nutné uviesť názov a cenu tarifu prípadne popis. Pri výpise týchto položiek bude okrem nastavovaných atribútov zobrazená informácia o počte zákazníkov s danou položkou ako v predošlej požiadavke.

Priorita: 4

Náročnosť: 2

3.1.2.6 Požiadavka č. 9

Účtovanie poplatkov za služby zákazníkom bude prebiehať na začiatku kalendárneho mesiaca a bude dvojakého typu. Zákazníkom ktorí vykonávajú ekonomickú činnosť (právnické osoby) budú generované faktúry a záznamy o platbe. Ostatným zákazníkom (fyzické osoby) budú generované len záznamy o platbe.

Záznam o platbe musí obsahovať nasledujúce atribúty:

- ID zákazníka
- dátum vytvorenia platby
- úhrada pre obdobie (rok/mesiac)
- suma k úhrade
- dátum úhrady
- uhradená suma
- fakturačný profil

Priorita: 4

Náročnosť: 3

3.1.2.7 Požiadavka č. 10

Faktúry budú generované výlučne do formátu pdf a budú obsahovať základné fakturačné údaje (dodávateľ, odberateľ, bankové spojenie, položky - názvy taríf definovaných v požiadavkách č. 7 a č.8., údaje pre platbu ako je VS, KS a výsledná suma k úhrade) pričom presné rozvrhnutie jednotlivých objektov nie je požadované.

Priorita: 3

Náročnosť: 4

3.1.2.8 Požiadavka č. 11

V súvislosti s predošlou požiadavkou, faktúry budú generované dynamicky na požiadanie (užívateľ klikne pre zobrazenie/stiahnutie faktúry) na základe záznamu o platbe doplnenou s požadovanými informáciami v požiadavke č.10. Faktúry sa budú posielat' výhradne elektronickou poštou a to v dobe generovania záznamu o platbe.

Priorita: 3

Náročnosť: 4

3.1.2.9 Požiadavka č. 12

Do budúcnosti sa ráta s podporou pre rôzne fakturačné profily pod ktorými budú evidovaní zákazníci a účtovanie ich poplatkov s využitím jednotného systému. Typickým príkladom je napr. predajca (reseller), ktorému bude možné nastaviť jednotnú zľavu z každého jeho pripojeného zákazníka a zaslaná spoločná faktúra za jeho pripojených zákazníkov.

Priorita: 3

Náročnosť: 4

3.1.2.10 Požiadavka č. 13

Užívateľ s dostatočnými oprávneniami bude môcť pridávať fakturačné profily pričom pri zakladaní uvedie či sa jedná o predajcu a nastaví mu nadradený fakturačný profil, pod ktorým bude predajca fakturovaný, prípadne zľavu.

Priorita: 3

Náročnosť: 3

3.1.2.11 Požiadavka č. 14

Implicitný fakturačný profil v systéme bude „NEFAKTUROVAT“. Zákazníkom s týmto profilom nebudú generované záznamy o platbe.

Priorita: 4

Náročnosť: 2

3.1.2.12 Požiadavka č. 15

Užívateľ systému s patričnými oprávneniami si bude vedieť pozrieť celkovú bilanciu platieb za ročné obdobie kde bude uvedené pre daný mesiac počet vygenerovaných platieb, počet uhradených, celková suma k úhrade, celkom zaplatené a informácia o tom koľko zostáva ešte k úhrade.

Užívateľovi bude poskytnutá možnosť zobrazit' detailný rozpis pre mesiac, ktorý bude obsahovať už konkrétne záznamy o platbách zákazníkov s možnosťou ich úhrady alebo popřípade stiahnuť faktúru.

Priorita: 4

Náročnosť: 4

3.1.2.13 Požiadavka č. 16

Systém bude okrem požiadavku č.15 poskytovať aj inú variantu prehľadu platieb, a to v podobe zoznamu zákazníkov (ID, meno, priezvisko, ročná bilancia, stavy pohľadávok v mesiacoch v roku) s možnosťou aplikovania filtru pre fakturačný profil, ročné obdobie, a stavu pohľadávok (zaplatené/nezaplatené). Pri výbere zákazníka sa užívateľovi zobrazia detaily o platbách a pohyby na účte zákazníka, kde bude taktiež mať možnosť manuálne uhradiť platbu.

Vo výpise kvôli väčšej prehľadnosti budú stavy pohľadávok / Nefakturované, Zaplatené, Zaplatené po splatnosti, Nezaplatené v splatnosti, Nezaplatené po splatnosti / rozlíšene farebne.

Priorita: 4

Náročnosť: 4

3.1.2.14 Požiadavka č. 17

Každému platiacemu zákazníkovi bude systém udržiavať jeho osobný účet, cez ktorý budú prechádzať všetky manuálne a automatické transakcie. Táto požiadavka je nevyhnutná pre lepšiu orientáciu s došlými platbami zákazníka.

Každý pohyb na jeho účte bude mať definované:

- fakturačný profil (na ktorý profil sa viaže platba)
- dátum pohybu
- z č. účtu zákazníka
- detail o platbe
- vs / ks / ss
- suma
- typ (napr. automatická prijatá platba, manuálne prijatá platba, automatická úhrada ...)

Priorita: 4

Náročnosť: 5

3.1.2.15 Požiadavka č. 18

Systém musí byť schopný importovať a spárovať platby zákazníkov podľa ich ID na osobné účty z bankového výpisu ktorý bude mať predom definovaný formát. Od systému sa očakáva aby správne zareagoval na nesprávne VS (zákazník sa nenachádza v systéme, alebo VS je špatne uvedený) alebo duplicitné platby pri importovaní výpisu za rovnaké obdobie. Každý import bude zaznamenaný a bude obsahovať

- dátum importu
- fakturačný profil
- názov importovaného súboru
- počet kladných obrátov (platieb)
- počet nespárovaných platieb
- zostáva skontrolovať
- login užívateľa kto importoval súbor

Priorita: 5

Náročnosť: 5

3.1.2.16 Požiadavka č. 19

V súvislosti s predchádzajúcou požiadavkou užívateľ bude vedieť manuálne prechádzať nespárované platby a bude môcť takúto platbu pripísať na osobný účet zákazníka vhodným spôsobom (rýchle vyhľadanie podľa kontaktných údajov, alebo ručne zadáním ID)

Priorita: 4

Náročnosť: 4

3.1.2.17 Požiadavka č. 20

Zaviesť systém lokalít pre rozdelenie aktívnych zariadení na základe ich fyzickej polohy v sieti. Lokalita bude obsahovať názov, ulicu, mesto a popri prípade údaje na kontaktnú osobu. Užívateľovi

budú poskytnuté informácie koľko zariadení a koncových staníc je v danej lokalite s možnosťou detailného zobrazenia. Taktiež bude vedieť pridávať/odstraňovať zariadenia do/z lokality.

Priorita: 2

Náročnosť: 2

3.1.2.18 Požiadavka č. 21

Spojením s požiadavkou č. 20, systém bude obsahovať a udržiavať stav o obsadenosti aktívnych zariadení na sieti pre danú lokalitu.

Priorita: 2

Náročnosť: 3

3.1.2.19 Požiadavka č. 22

V systéme budú definované rozsahy IP adries pre koncových zákazníkov.

Pre pridanie nového rozsahu bude musieť užívateľ uviesť názov, IP adresu siete, masku siete a bránu.

Od systému sa očakáva aby neumožnil priradiť viacerým klientom rovnakú IP adresu. Taktiež bude zobrazovať počet použitých a počet voľných IP adries v danom rozsahu.

Priorita: 4

Náročnosť: 4

3.1.2.20 Požiadavka č. 23

Pri registrácii zákazníka a prípojky bude systém očakávať:

- kontaktné údaje (meno, priezvisko, adresu, tel, email)
- v prípade právnickej osoby i názov organizácie, IČO, DIČ, IČDPH
- typ prípojky (wifi alebo káblové pripojenie)
- názov lokality
- aktívne zariadenie do ktorého je koncová stanica pripojená
- v prípade switchu uviesť i číslo portu
- MAC adresu
- lokálnu IP adresu (možnosť výberu z preddefinovaných IP rozsahov)
- verejnú IP adresu (možnosť výberu z preddefinovaných IP rozsahov)
- adresu prípojky ak je rozdielna od kontaktnej adresy
- fakturačný profil
- názov tarify
- inštalčný poplatok
- parametre pre zmluvu (viazanosť, automatické predĺženie...)
- platby od / do dátumu
- stav prípojky (aktívna alebo čakajúca na pripojenie)
- dátum aktivácie (v prípade aktivácie)
- poprípade poznámky

Priorita: 4

Náročnosť: 3

3.1.2.21 Požiadavka č. 24

Automatické prevedenie nevyhnutných akcií, aby bola prípojka aktívna do siete internet po dokončení registrácie so stavom aktívna. Aktiváciu prípojky bude možné odložiť na dobu neurčitú označím stavu ako „čakajúca na pripojenie“.

Priorita: 5

Náročnosť: 4

3.1.2.22 Požiadavka č. 25

Zoznam prípojok čakajúcich na aktiváciu bude ľahko dostupný a bude obsahovať kontaktnú osobu, tel. kontakt a adresu prípojky. Užívateľovi bude ponúknutá voľba pre dokončenie registrácie s vyplneným formulárom vyplývajúcim z požiadavky č. 23, kde bude mať ešte možnosť zmeniť technické a zmluvné parametre prípojky. Užívateľ v tomto kroku môže aj zamietnuť aktiváciu z rôznych dôvodov pričom kontakt na zákazníka a informácia o potencionálnej prípojke je udržiavaná v systéme.

Priorita: 4

Náročnosť: 3

3.1.2.23 Požiadavka č. 26

Po aktivácii prípojky a s podmienkou že má nastavený fakturačný profil, ktorý nie je označený ako predajca alebo „NEFAKTUROVAT“, systém automaticky vytvorí podklady pre zmluvu zo zadaných údajov. Užívateľ ďalej bude môcť túto zmluvu editovať a následne stiahnuť vo formáte pdf.

V systéme bude evidované:

- dátum a čas vytvorenia zmluvy
- login kto zmluvu vytvoril
- platná od/do (dátum)
- poslaná dňa, login kto ju poslal (označil v systéme)
- vrátená dňa, login kto ju prevzal (označil v systéme)

Priorita: 2

Náročnosť: 2

3.1.2.24 Požiadavka č. 27

V súvislosti s predošlou požiadavkou systém musí umožniť nastavenia dátumov odoslania a vrátenia zmluvy v danom poradí. Systém bude zabezpečovať automatickú kontrolu dátumov. Pri nenastavení dátumu doručenia po stanovenej dobe 14 dní od dňa odoslania, bude prípojka zákazníkovi obmedzená spôsobom blokovania portu 80 (web) a zobrazená informácie o dôvode blokovania.

Priorita: 1

Náročnosť: 4

3.1.2.25 Požiadavka č. 28

Užívateľ bude v systéme vedieť zadávať skupiny IP adries a doménových mien pre blokovanie webových stránok pričom implicitná skupina bude „Všetko“. Následne bude vedieť tieto skupiny aplikovať na konkrétnu prípojku s vymedzením dátumov od kedy do kedy je toto blokovanie platné a

určením typu oznamu, ktorý bude zobrazený zákazníkovi. Blokovanie cieľových stránok bude umožnené aj jednotlivým zapísaním adresy/adries bez nutnosti vytvárať skupinu.

Priorita: 1

Náročnosť: 4

3.1.2.26 Požiadavka č. 29

Systém implicitne musí obsahovať 2 typy hlášok pre automatické blokovania. (nevrátenie zmluvy, neplatenie záväzkov). Užívateľ si bude môcť pridávať ďalšie hlášky pre explicitné blokovania stránok.

Priorita: 1

Náročnosť: 1

3.1.2.27 Požiadavka č. 30

V systéme sa predpokladajú dva typy vyhľadávania zákazníkov. Prvý typ vyhľadávania známy ako „autosuggestion“ bude zameraný na rýchly prístup (s vhodným umiestnením vyhľadávacieho poľa) z ktoréhokoľvek modulu systému a bude vyhľadávať spomedzi nasledujúcich atribútov:

- meno alebo priezvisko
- ulica, mesto
- tel. alebo e-mail kontakt
- názov organizácie alebo IČO
- IP adresy (privátna alebo verejná)

Druhý spôsob vyhľadávania bude vyhľadávať taktiež fulltextovo nad hore uvedenými atribútami ale navyše bude umožňovať aplikovania rôznych filtrov a to vyhľadávanie podľa:

- fakturačného profilu
- typu prípojky (wifi alebo káblová)
- stavu prípojky (aktívna, čakajúca, vymazaná, bloková)
- internetovej tarify
- iné detaily (napr. Prípojky s verejnou IP adresou, nevrátenou zmluvou a i)

Po vyhľadaní sa zobrazí zoznam ktorý splňuje dané kritéria a bude obsahovať:

- ID zákazníka
- meno a priezvisko resp. Názov organizácie
- adresu zákazníka
- stav účtu

V oboch prípadoch po označení zákazníka sa zobrazí jeho kontakt, zoznam prípojok, zoznam zmlúv, záznamy o platbách za aktuálny rok a posledných 10 transakcií na jeho osobnom účte.

Priorita: 3

Náročnosť: 3

3.1.2.28 Požiadavka č. 31

V zozname prípojok budú uvedené základné informácie

- stav prípojky

- typ prípojky
- lokálna a verejná IP adresa
- názov lokality
- názov tarify

Užívateľ si bude môcť zobraziť detaily konkrétnej prípojky

Priorita: 3

Náročnosť: 2

3.1.2.29 Požiadavka č. 32

Detail prípojky okrem hore uvedených atribútov musí navyše obsahovať

- MAC adresu
- šírka pásma download i upload
- konkrétny aktívny prvok v prípade prepínaču i port na ktorom je prípojka zapojená
- loginy a časové údaje (vytvorenie prípojky, aktivácia, posledná modifikácia)
- fakturačný profil
- platby od/do

Okrem spomenutých základných údajov a umiestnenia prípojky bude tu uvedená história a zoznam aktívnych blokovanie stránok s možnosťou deaktivácie.

Priorita: 3

Náročnosť: 3

3.1.2.30 Požiadavka č. 33

Zavedenie počítania prenesených dát smerom od/k zákazníkovi. Vzorkovanie sa predpokladá s minimálnou periódou 5minút. Systém bude musieť byť schopný okrem zobrazenia konkrétnych hodnôt reprezentovať namerané data formou grafov a taktiež určiť priemernú rýchlosť za zvolené obdobie.

Priorita: 4

Náročnosť: 5

3.1.2.31 Požiadavka č. 34

So zavedením p.č. 33 zobrazovať namerané hodnoty globálne pre všetkých zákazníkov a to v časových horizontoch:

- posledná hodina
- za celý deň
- poslený týždeň
- posledný mesiac
- posledný rok
- celkovo

s možným použitím filtrov (napr. typ prípojky (wifi,kabel), lokalita) a s určením vypísaním posledných n klientov, pričom budú zoradený od najviac prenesených dát tzv. „Top n“

Tieto štatistiky sa budú dať pozrieť i pre konkrétnu prípojku v detailoch o prípojke.

Priorita: 4

Náročnosť: 3

3.1.2.32 Požiadavka č. 35

Do budúcnosti počítať s požiadavkou pre integráciu aktívnych prvkov siete (prepínače, prístupové body) a komunikáciu s nimi. Napr. na základe p.č. 23 aplikovanie port-security na prepínači pre port.

Priorita: 3

Náročnosť: 4

3.1.3 Rozhranie systému

3.1.3.1 Požiadavka č. 36

Celý systém bude riadený pomocou jednotného grafického užívateľského rozhrania, pričom nie sú kladené špecifické požiadavky na jeho vzhľad a rozhranie. Systém bude komunikovať s užívateľom v anglickom jazyku a v prípade potreby, možnosť rýchlej implementácii rôznej jazykovej mutácii.

Priorita: 2

Náročnosť: 2

3.1.3.2 Požiadavka č. 37

Realizovať grafické užívateľské rozhranie formou web stránky zobraziteľnej v ľubovoľnom internetovom prehliadači.

Priorita: 2

Náročnosť: 3

3.1.4 Zdroje

3.1.4.1 Požiadavka č. 38

Systém musí byť vytvorený s využitím voľne dostupných technológií a prostriedkov za účelom maximálneho zníženia nákladov na jeho vývoj a údržbu. Požiadavky na využitie konkrétnej technológie nie sú špecifikované.

Priorita: 2

Náročnosť: 4

3.1.4.2 Požiadavka č. 39

Minimalizovať požiadavky na vybavenie užívateľských počítačov a eliminovať tak potrebu inštalovať dodatočný SW za účelom správy a používania systému.

Priorita: 1

Náročnosť: 1

3.1.4.3 Požiadavka č. 40

Od systému sa predpokladá, že jeho rozhranie bude realizované na inom stroji (web serveri) než je hlavná brána ISP. Komunikácia medzi systémom a bránou nebola špecifikovaná.

Priorita: 3
Náročnosť: 5

3.1.5 Dokumentácia

3.1.5.1 Požiadavka č. 41

Vytvorenie užívateľskej príručky obsahujúcej popis užívateľského grafického rozhrania. Súčasťou dokumentácie budú i príklady použitia systému a vysvetlené princípy práce so systémom.

Priorita: 2
Náročnosť: 3

3.1.5.2 Požiadavka č. 42

Za účelom sprehľadnenia a uľahčenia prípadných úprav jadra systému vygenerovať programátorskú dokumentáciu s pomocou dokumentačných nástrojov danej technológie.

Priorita: 1
Náročnosť: 2

3.2 Use case model

Pre zobrazenie funkčnej štruktúry systému z pohľadu užívateľa bol zvolený use case model. Tento model (viď príloha č. 2) definuje správanie systému a tvorí tak grafické zobrazenie analýzy funkcionálnych požiadaviek. Nasledovné podkapitoly dopĺňujú a upresňujú zobrazenú hierarchiu aktérov a kľúčové prípady použitia.

3.2.1 Aktéri

3.2.1.1 Technik

Užívateľ systému s najnižšími právami. Užívatelia tohoto typu môžu pridávať – aktivovať zákazníkov a ich prípojky len pokiaľ sú v stave „čakateľ na pripojenie“. Nie je im poskytnutý podrobný zoznam registrovaných zákazníkov (napr. stav účtu zákazníka) iba vyhľadávanie definované v požiadavke č.30 prvý odstavec a zoznam čakateľov p.č. 25. Užívateľ má do ostatných modulov systému zamedzený prístup.

3.2.1.2 Admin

Má prístup takmer do všetkých častí systému, pričom niektoré moduly sú obmedzené len na „read-only“ (zobrazenie a prezeranie).

3.2.1.3 Super Admin

Užívateľ s najvyššími prístupovými právami. Má prístup do celého systému a kontrolu nad všetkými evidenciami vrátane financií.

3.2.2 Prípady použitia

V jednotlivých scenároch prípadov použitia vystupujú aktéri ako oprávnený užívateľ systému (ďalej len „užívateľ“) podľa use case modelu.

3.2.2.1 Registrácia zákazníka

Pridávanie nových zákazníkov do systému je umožnené užívateľom s právami „Admin“ a „Super Admin“

Scenár prípadu použitia

- **Registrácia nového zákazníka**
 1. užívateľ iniciuje pridanie nového zákazníka
 2. systém bude požadovať informácie o novom zákazníkovi a parametroch prípojky zadaných v p.č. 23
 3. užívateľ zadá požadované informácie a iniciuje ich uloženie
 4. systém skontroluje zadané informácie a v prípade úspechu zaregistruje zákazníka a zobrazí ich
 5. v prípade chyby systém zobrazí túto skutočnosť s možnosťou nápravy
 6. systém uloží informáciu o pridaní zákazníka do systémového logu

3.2.2.2 Nájsť zákazníka

Vzhľadom k danej analýze požiadaviek je tento prípad zovšeobecnením vyhľadávania zákazníka podľa kontaktných údajov a registrovanej prípojky.

Scenár prípadu použitia

- **Nájsť kontakt**
 1. užívateľ iniciuje vyhľadanie zákazníka zadaním jeho kontaktných údajov
 2. systém vráti výsledok hľadania formou zoznamu s priamymi odkazmi na zákazníka
- **Nájsť prípojku**
 1. užívateľ iniciuje vyhľadanie zákazníka zadaním jeho IP adresy prípojky
 2. systém vráti výsledok hľadania formou zoznamu s priamymi odkazmi na prípojky zákazníka

3.2.2.3 Aktivácia prípojky

Užívateľ, najčastejšie technik, môže aktivovať prípojku zákazníka, ktorý je už v systéme predom zaregistrovaný (má stav čakateľ na pripojenie). Aktivácia prípojky môže byť taktiež učená hneď pri registrácii užívateľom s patričným oprávnením.

Scenár prípadu použitia

- **Aktivácia prípojky**

1. užívateľ vyhľadá zákazníka podľa vyššie uvedeného scenára alebo vyberie zákazníka zo zoznamu čakateľov na pripojenie
2. užívateľ iniciuje aktiváciu prípojky
3. systém zobrazí predvyplnený formulár s údajmi pri registrácii
4. užívateľ môže zmeniť technické parametre prípojky (napr. prideliť IP adresu)
5. užívateľ dokončí registráciu so stavom prípojky „aktívna“ alebo „vymazaná“
6. systém skontroluje zadané informácie a v prípade úspechu zobrazí detail zákazníka
7. systém uloží informáciu o aktivácii do systémového logu

3.2.2.4 Zmeniť technické parametre prípojky

Technickými parametrami sa rozumie predovšetkým MAC adresa, privátna či verejná IP adresa, lokalita a typ prípojky

Scenár prípadu použitia

- **Zmeniť technické parametre prípojky**

1. užívateľ vyhľadá zákazníka
2. užívateľ iniciuje zmenu konkrétnej prípojky
3. systém zobrazí formulár s aktuálnymi parametrami
4. užívateľ vykoná potrebné zmeny
5. systém aktualizuje zmeny a uloží informáciu o zmene do systémového logu

3.2.2.5 Štatistiky zákazníkov

Štatistiky prenesených dát smerom k a od zákazníka tvoria taktiež podstatnú časť systému. Oprávneným užívateľom umožňuje prezerať rôzne typy štatistík a grafov s aplikovaním požadovaných filtrov.

Scenár prípadu použitia

- **Zobraziť celkové štatistiky**

1. užívateľ iniciuje zobrazenie štatistík
2. systém zobrazí posledných 10 záznamov so štatistikami zákazníkov a ich prípojok za poslednú hodinu zoradených zostupne spolu s priamym odkazom na detailné štatistiky konkrétnej prípojky
3. užívateľ môže aplikovať rôzne filtre (napr. Zobrazenie štatistík za posledný mesiac) alebo zmeniť počet zobrazovaných záznamov
4. po aplikovaní filtrov systém znova zobrazí požadované štatistiky

- **Zobraziť detailné štatistiky zákazníka**

1. užívateľ vyhľadá zákazníka a jeho prípojku podľa scenára popísaného v 3.2.2.2. alebo využije priamy odkaz z predošlého scenára prípadu použitia
2. systém umožní prezeranie podrobných štatistík prípojky
3. užívateľ môže vyžiadať grafickú reprezentáciu štatistík
4. systém zobrazí grafy vytvárenosti prípojky

3.2.2.6 Zmeniť údaje zákazníka

Umožňuje zmeniť kontaktné údaje zákazníka.

Scenár prípadu použitia

- **Zmeniť údaje zákazníka**

1. užívateľ vyhľadá alebo vyberie zákazníka zo zoznamu zákazníkov
2. systém zobrazí jeho aktuálne údaje
3. užívateľ iniciuje zmenu údajov
4. systém zobrazí editačný formulár s možnosťou opravy údajov
5. užívateľ potvrdí vykonané zmeny
6. systém skontroluje zadané informácie a v prípade úspechu prepíše aktuálne údaje
7. v prípade neúspechu zobrazí užívateľovi správu o chybe s možnosťou opravy
8. systém uloží informáciu o zmene do systémového logu

3.2.2.7 Zobrazit' blokovanie prístupu zákazníka

Zobrazuje zoznam aktuálnych i históriu blokovaní web stránok na základe p.č. 28. Predpokladá sa možnosť akcie užívateľa aby dané pravidlo blokovania mohol zo zoznamu rýchlim spôsobom deaktivovať resp. pridať nové pravidlo(á)

Scenár prípadu použitia

- **Zobrazit' blokovanie prístupu zákazníka**

1. užívateľ vyhľadá zákazníka a jeho prípojku podľa scenára popísaného v 3.2.2.2.
2. užívateľ iniciuje zobrazenie blokovaní prístupu pre konkrétnu prípojku zákazníka
3. systém zobrazí zoznam blokovaní a ponúkne požadované akcie

- **Pridať nové blokovanie**

1. užívateľ iniciuje zadanie nového blokovaní
2. systém zobrazí formulár a bude požadovať informácie o blokovaní
3. užívateľ vyberie/zadá rozsah, dôvod a čas blokovaní služieb www
4. systém uloží zadané informácie a aktivuje blokovanie
5. systém uloží informáciu o pridaní blokovaní do systémového logu

- **Zrušiť blokovanie**

1. užívateľ iniciuje zrušenie aktívneho blokovaní zo zoznamu
2. systém nastaví dátum ukončenia a zároveň deaktivuje blokovanie
3. systém uloží informáciu o zrušení do systémového logu

3.2.2.8 **Zobraziť služby zákazníka**

Služby zákazníka umožňujú účtovať poplatky za využívanie dodatočných služieb nezávisle od skutočného stavu a viaže sa na konkrétnu prípojku. Tak ako predošlý prípad použitia umožňuje i tento zobraziť zoznam aktuálnych i v minulosti využívaných služieb. Predpokladá sa možnosť akcie užívateľa pridávať a uberať služby zo zoznamu rýchlym spôsobom.

Scenár prípadu použitia

- **Zobraziť služby zákazníka**
 1. užívateľ vyhľadá zákazníka a jeho prípojku podľa scenára popísaného v 3.2.2.2.
 2. užívateľ iniciuje zobrazenie služieb zákazníka pre konkrétnu prípojku
 3. systém zobrazí zoznam využívaných služieb a ponúkne požadované akcie
- **Nová služba**
 1. užívateľ iniciuje pridanie novej služby
 2. systém zobrazí formulár a zoznam dostupných služieb
 3. užívateľ vyberie požadovanú službu a nastaví dátumy od/do kedy bude služba spoplatňovaná
 4. systém uloží zadané informácie a znamená túto skutočnosť do systémového logu
- **Zrušiť službu**
 1. užívateľ iniciuje zrušenie aktívnej služby zo zoznamu
 2. systém nastaví dátum ukončenia pre spoplatňovanie služby
 3. systém uloží informáciu o zrušení služby do systémového logu

3.2.2.9 **Zmluvy zákazníka**

Užívateľ môže v tomto prípade použitia pracovať so zmluvami zákazníka. Od systému sa predpokladá, že zmluvy budú predvyplnené automaticky pri registrácii zákazníka s prípojkou.

Scenár prípadu použitia

- **Zmluvy zákazníka**
 1. užívateľ vyhľadá alebo vyberie zákazníka zo zoznamu zákazníkov
 2. systém okrem detailov zákazníka zobrazí zoznam zmlúv a dodatkov
 3. užívateľovi sú ponúknuté rýchle voľby: editovať zmluvu, nastaviť dátum odoslania alebo vrátenia zmluvy, stiahnutie zmluvy
- **Editovať zmluvu**
 1. užívateľ iniciuje editovanie konkrétnej zmluvy zo zoznamu
 2. systém zobrazí predvyplnený formulár s údajmi zadanými pri registrácii zákazníka a prípojky
 3. užívateľ zmení požadované informácie a iniciuje uloženie zmien
 4. systém skontroluje dané zmeny a v prípade úspechu ich uloží
 5. v prípade neúspechu zobrazí užívateľovi správu o chybe s možnosťou opravy
 6. systém uloží informáciu o zmene do systémového logu

3.2.2.10 Import platieb

Podstatnou časťou celého systému je účtovanie poplatkov zákazníkom za poskytovanie internetového pripojenia a služieb. Import platieb bude spracúvať súbor obsahujúci výpis z účtu ktorý bude mať preddefinovaný formát.

Scenár prípadu použitia

- **Import platieb**
 1. užívateľ iniciuje importovanie platieb
 2. systém bude od užívateľa očakávať výber fakturačného profilu a súbor s platbami
 3. užívateľ vyberie profil a vloží požadovaný súbor
 4. systém automaticky spáruje platby a o výsledku bude informovať užívateľa
 5. systém uloží informácie o tejto akcii do systémového logu

4 Implementácia systému

4.1 Realizácia kľúčových častí systému

Cieľom tejto kapitoly je ukázať čitateľovi možnosť realizácie daného projektu na základe analýzy požiadaviek v kapitole 3.1 a use case modelom v kapitole 3.2. Funkcionalita systému bola dostatočne špecifikovaná a podľa teoretických základov popísaných v kapitole 2.2.1. a 2.2.2. nebol problém vytvoriť detailný návrh systému s požadovanými vlastnosťami. Pre bližšie priblíženie realizácie systému bude v nasledovných podkapitolách realizácia konkrétnych častí demonštrovaná na príkladoch.

Realizácia bola rozdelená na dve časti (výkonná a web rozhranie). Hlavná časť ako už názov napovedá je výkonná, ktorá obsahuje sadu skriptov zabezpečujúcich interpretáciu užívateľských podnetov ale i automatických akcií na nastavovanie firewallu a QoS na strane smerovača. Jednotlivé akcie užívateľ zadáva pomocou web rozhrania. Táto časť bola spolu s databázou nainštalovaná na inom stroji ako je smerovač, aby sa predišlo k zbytočnej záťaži. Komunikácia medzi týmito dvomi časťami systému prebieha pomocou nastavovaním príznakov vedúcich k určitej akcii. Skript na strane hlavnej brány kontroluje tieto príznaky s vhodne zvolenou periódou pomocou plánovača úloh cron⁷. Vo všeobecnosti jednotlivé príznaky môžu nadobúdať 3 stavy a ich popis je uvedený v Tabuľke č. 6.

- stav 0 – systém je v prevádzke, nie sú žiadne nové zmeny
- stav 1 – systém pri nasledujúcej kontrole vykoná požadované zmeny
- stav 2 – avizovaná zmena stavu, po zvolenej perióde prejde do stavu 1. tento stav slúži k zabráneniu opakovania akcií pri dlhšej práci so systémom ktoré by viedli k častým, zbytočným reštartom služieb na strane smerovača.

Príznak	Popis
update_qos_conf	Skript kontrolujúci stav príznakov pri najbližšej kontrole vygeneruje na základe nastavení taríf za internet a profilov pre QoS nový konfiguračný súbor.
update_system	Vygeneruje nový hosts súbor pre QoS
restart_qos	Reštartuje QoS
restart_fw	Reštartuje firewall

Tabuľka 6: Popis príznakov

⁷ Je v OS Linux systémový proces, ktorý automatizované v určitý čas spúšťa skripty resp. procesy.

4.1.1 Registrácia a aktivácia prípojky

Registrácia zákazníka spolu s prípojkou prebieha cez web rozhranie na základe podnetu užívateľa. Pre aktiváciu prípojky sú z technického hľadiska požadované dva parametre: IP adresa prípojky a pridelenie šírky pásma (tarify).

4.1.1.1 Internetové tarify – definícia šírky pásma a profilov pre QoS

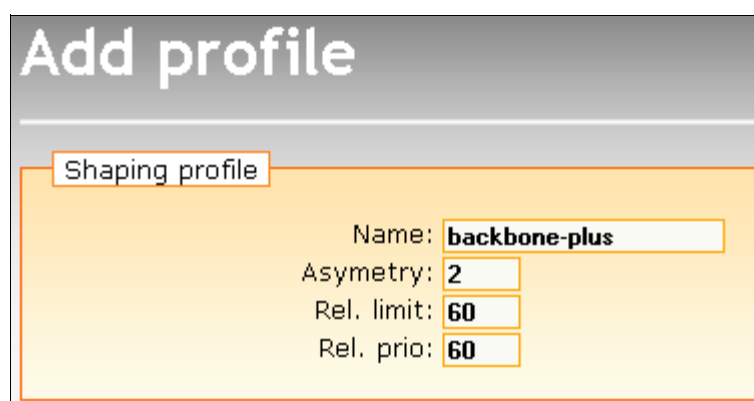
Systém umožňuje jednoduchú správu kľúčových slov pre QoS (viď kap. 2.2.2) a parametrov pre aplikovanie FUP zavedením tzv. shapovacích profilov. Jednotlivé profily sú tak podkladom pre vytváranie internetových taríf kde na základe nich zákazník platí a je mu pridelená šírka pásma. Nasledujúci príklad demonštruje pridanie nového profilu a tarify.

Príklad:

Potrebujeme vytvoriť internetovú tarifu s aplikovaním FUP pre zákazníkov s názvom „Internet mini“ ktorá bude poskytovať zákazníkovi max. šírku pásma download/upload 2048/1024 kbps, garantovanú šírku pásma 128/64 kbps a bude spoplatnená sumou 16,27 €.

Na základne tohto príkladu je potrebné mať vytvorený (alebo vytvoriť viď Obrázok č. 2) profil pre QoS, ktorý bude definovať asymetrický pomer download/upload „2“ a parametre pre aplikovanie FUP ktoré sú relatívne voči garantovanej rýchlosti. V našom prípade keď nastavíme konštantu „Rel.limit“ resp. „Rel.prio“ na hodnotu „60“ povedie pri garantovanej rýchlosti 128 kbps nastavením limitu pre objem prenesených dát (download+upload) na hodnotu $128 \times 60 = 7680$ MB. Po prekročení tohto limitu bude zákazníkovi znížená šírka pásma na jeho garantovanú (v našom prípade 128 kbps) a taktiež znížená priorita obsluhy. Aplikovanie FUP je možné vypnúť zadáním hodnoty „0“ pre spomínané konštanty.

Pri manipulácii so shapovacími profilmi systém nastavuje príznak `update_qos_conf`.



The screenshot shows a web form titled "Add profile". Below the title is a section labeled "Shaping profile" which contains four input fields:

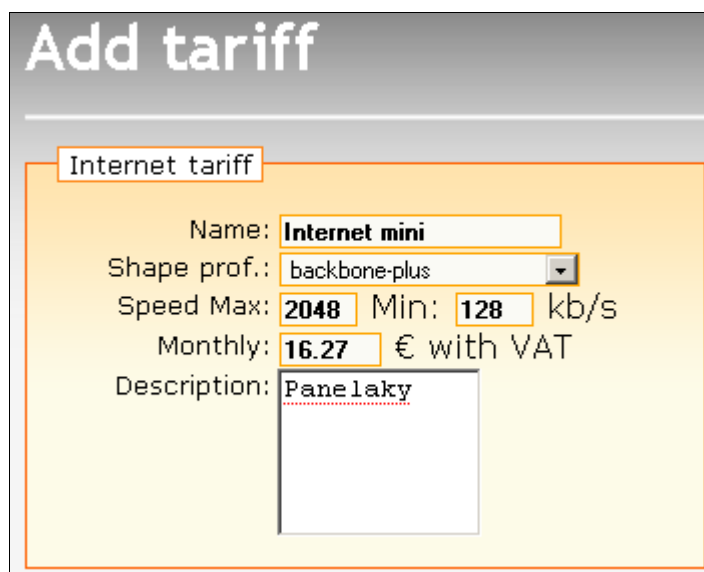
- Name: **backbone-plus**
- Asymetry: **2**
- Rel. limit: **60**
- Rel. prio: **60**

Obrázok 2: Web rozhranie: Vytvorenie profilu pre QoS

Po úspešnom vytvorení profilu pre QoS, skript bežiaci na strane smerovača zareaguje na zmenený príznak a vytvorí nový konfiguračný súbor pre QoS, kde podľa parametrov z Obrázku č.2 bude mimo iného obsahovať i nasledujúce riadky (popis jednotlivých parametrov je v kap.2.2.2.). Skript po kontrole nastaví príznak opäť na hodnotu „0“

```
keyword backbone-plus
backbone-plus-asymetry-ratio 2
backbone-plus-asymetry-treshold 0
backbone-plus-magic-relative-limit 60
backbone-plus-magic-relative-prio 60
backbone-plus-magic-fixed-limit 0
backbone-plus-magic-fixed-prio 0
backbone-plus-htb-default-prio 1
```

Pridávanie a modifikácia taríf za internet je cez web rozhranie veľmi flexibilná a umožňuje interaktívnu zmenu šírky pásma i stávajúcim zákazníkom, ktorí majú pridelenú danú tarifu. Operácie nad tarifikami nastavujú príznak „update_system“ a podieľajú sa tak na zmene súboru „hosts“ ktorý využíva program pre zabezpečenie QoS. Obrázok č. 3 definuje požadovanú tarifu podľa príkladu.



The screenshot shows a web form titled "Add tariff". Inside, there is a section labeled "Internet tariff". The form contains the following fields and values:

- Name:
- Shape prof.:
- Speed Max: Min: kb/s
- Monthly: € with VAT
- Description:

Obrázok 3: Web rozhranie : Pridanie novej tarify

4.1.1.2 Aktivácia prípojky – prístup do siete internet

Predpokladajme že na klientskej stanici resp. zariadení (ďalej len „klient“) sú nastavené potrebné informácie k pripojeniu siete internet (IP adresa, maska, východzia brána a i) pomocou ručnej konfigurácie alebo DHCP serverom. Na to aby mohol klient komunikovať so sieťou internet je nevyhnutné prípojku aktivovať a urobiť tak patričné kroky na bráne. Proces aktivácie klientskej prípojky cez predstavovaný systém prebieha automaticky a je plne automatizovaný počnúc zaregistrovaním v IS a označením stavu ako „aktívna“. Po úspešnej registrácii je nastavený príznak *update_system* a *restart_fw* na hodnotu 3 .

Príklad

Pokračovanie príkladu z kapitoly 4.1.1.:

Zákazníkovi bola pridelená IP adresa v rámci lokálnej siete 10.100.0.13. Zákazník však na klientskej stanici požaduje i verejnú IP adresu a rýchlosť zodpovedajúcej internetovej tarify „Internet mini“. V tomto momente máme všetky potrebné informácie aby mohla byť prípojka zaregistrovaná. V IS v procese registrácie v časti „pripojenie“ (viď Obrázok č.4) nastavíme detaily o prípojke zákazníka. Aby sa predošlo k prípadným chybám, IP adresy sa nedajú ručne zadávať ale sa vyberú z nadefinovaného rozsahu podľa p.č. 22. Následne v časti „zmluva“ sa zvolí požadovaná tarifa. Pre úspešnú registráciu je potrebné okrem parametrov uvedených na obrázku č. 4 uviesť kontakt na zákazníka (meno, priezvisko, adresa, kontakt).

Podľa tohto príkladu skript bežiaci na strane smerovača vygeneruje nasledovné riadky pre pravidlá do firewallu:

```
iptables -A FORWARD -d 10.100.0.13 -j ACCEPT
iptables -A FORWARD -s 10.100.0.13 -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.100.0.13 -j SNAT --to 87.244.192.55
iptables -t nat -A PREROUTING -d 87.244.192.55 -j DNAT --to-destination 10.100.0.13
```

a vygeneruje nový súbor „hosts“ pre zabezpečenie QoS, ktorý bude obsahovať nasledovný riadok:

```
10.100.0.13 10.100.0.13 #backbone-plus-128-2048
```

Connection	Contract
Type: Cable	Invc group: SURANY.net
Location: Lucna_3	Tariff: Internet mini
Switch: SW_Lucna7	Speed Max: 2048 Min: 128 kb/s
Port: 3	Asymetry: 2 up=down/ratio
MAC:	Monthly: 16.27 € with tax
Local IP: 10.100.0.13	Installation: € with tax
Public IP: 87.244.192.55	Termination: <input type="checkbox"/> With allegiance
Address: <input checked="" type="checkbox"/> Same as contact	<input type="text"/> months
Street:	Prolongation: <input checked="" type="radio"/> No <input type="radio"/> Yes
City:	Payment from: 2009-05-01
PSC:	Payment to:

Obrázok 4: Web rozhranie: Nastavenie parametrov prípojky

4.1.2 Blokovanie prístupu

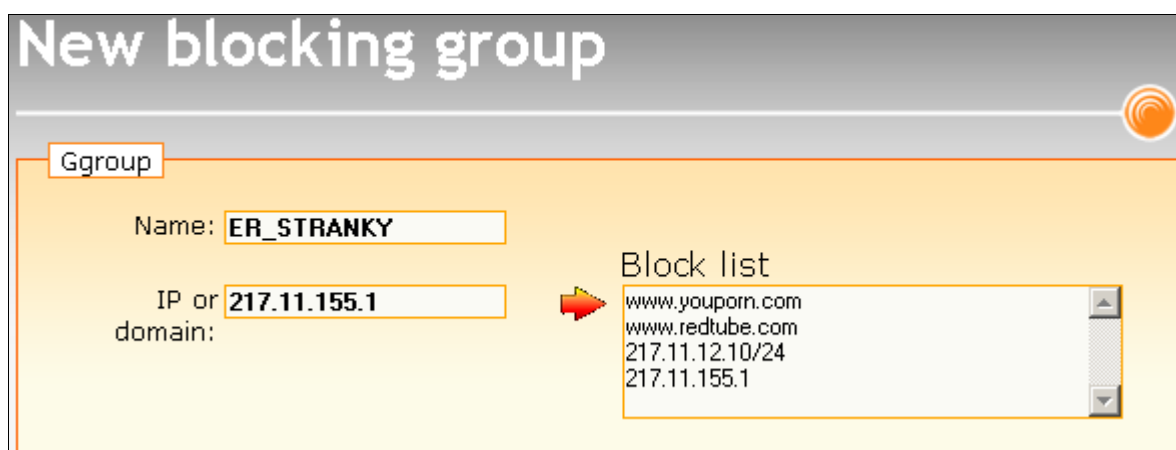
Ďalšou kľúčovou časťou systému je buď čiastočné alebo úplné obmedzenie služieb www (blokovanie portu 80) podľa p.č. 27 až 29 s následným presmerovaním na lokálny web server kde je zákazníkovi zobrazená informácia o dôvode blokovania. Ako web server by mal byť zvolený na lokálnej sieti kvôli identifikácii vzdialeného klienta. Klient je identifikovaný podľa lokálnej IP adresy a je mu zobrazený text (podľa nastavení v blokovani prípojky vid' Obrázok č. 7) pri pokuse zobrazit' požadovanú web stránku. Tento mechanizmus je realizovaný pomocou firewallu iptables. (Vid' spracovanie paketu firewallom iptables v kap. 2.2.1.1.) Keďže chceme menit' cieľové adresy na náš web server musíme pakety spracovať ešte pred procesom smerovania paketu v pravidle PREROUTING a použiť ako cieľ DNAT. Štýl zápisu bude zrejmý z nasledovného príkladu.

Príklad:

Požadujeme aby klient s IP adresou 10.100.0.13 mal blokované všetky web stránky a pokusy o prístup na ne boli presmerované na náš web server ktorý má IP 10.10.254.254 a počúva na porte 8080. Výsledné pravidlo bude vyzerat' nasledovne:

```
iptables -I PREROUTING -t nat -s 10.100.0.13 -d 0/0 -p tcp --dport 80 -j  
DNAT --to 10.10.254.254:8080
```

Ako je vidno zo zápisu správa takýchto blokovaní by bola ručným zadávaním pravidiel značne komplikovaná a neflexibilná. Preto predstavovaný systém implementuje správu blokovaní ľahko dostupnou cez web rozhranie, kde je možné definovať skupiny blokovanych cieľových adries (Obrázok č.5) a užívateľom definovaných správ o dôvode blokovania (Obrázok č. 6). Systém ponúka dva typy blokovaní ktoré sú popísané v nasledujúcich podkapitolách. Oba typy pri aplikovaní alebo zmene blokovania nastavujú príznak *restart_fw* na hodnotu 3.



Obrázok 5: Web rozhranie: Vytvorenie blokovacej skupiny

4.1.2.1 Manuálne blokovanie

Manuálne blokovanie aplikuje a je plne pod kontrolou užívateľa cez web rozhranie systému. Pre aplikovanie blokovania je nutné mať vytvorenú informáciu o dôvode blokovania (Obrázok č.6), ktorá bude zobrazená zákazníkovi, ďalej je potrebné uviesť adresu resp. doménu ktorá bude blokováná a vymedziť dátumy kedy je blokovanie aktívne (viď obrázok č. 7. poznámka: v čase písania tohto textu aplikovanie blokovania na jednotlivé dni v týždni nebolo implementované). Ak by bolo potrebné zadať viac adries než jednu, je možné vytvoriť si skupinu týchto adries (Obrázok č.5) Adresu blokovania je možné zadávať 3 spôsobmi:

- IP adresu cieľovej stránky (napr. 217.22.11.1)
- sieť s maskou (napr. 212.22.1.2/24)
- doménové meno (napr. www.domena.sk)

V prípade zadania skupiny, pole pre zadávanie adresy slúži pre vylúčenie adresy z definovanej skupiny. Po vytvorení blokovania na konkrétnu prípojku sa na strane routera vygenerujú pravidlá pre firewall ako je ukázané na príklade v kap. 4.1.2

Systém implicitne obsahuje blokovaciu skupinu „All“ pod ktorú spadajú všetky adresy (0/0) a dve informácie o dôvode určené pre automatické blokovanie ktoré sú popísané v kap 4.1.2.2.

Obrázok 6: Web rozhranie: Vytvorenie správy o dôvode zablokovania

Day	Time Range
Mon:	<input checked="" type="checkbox"/> from 0 to 24 hod
Tue:	<input checked="" type="checkbox"/> from 0 to 24 hod
Wed:	<input checked="" type="checkbox"/> from 0 to 24 hod
Thu:	<input checked="" type="checkbox"/> from 0 to 24 hod
Fri:	<input checked="" type="checkbox"/> from 0 to 24 hod
Sat:	<input checked="" type="checkbox"/> from 0 to 24 hod
Sun:	<input checked="" type="checkbox"/> from 0 to 24 hod

Obrázok 7: Web rozhranie: Aplikovanie blokovania na prípojku zákazníka

4.1.2.2 Automatické blokovanie

Väčšinou má blokovanie určitý špecifický význam. V tomto projekte je práve požadovaný automatický typ blokovania, ktorý mimo iného plní funkciu upomienky, ktorá je vyžadovaná zo zákona pred odpojením zákazníka. Dá sa predpokladať, že väčšina pripojených zákazníkov používa práve služby www, takže tento spôsob upozornenia je veľmi efektívny a uprednostňovaný pred papierovou resp. upomienkou cez email.

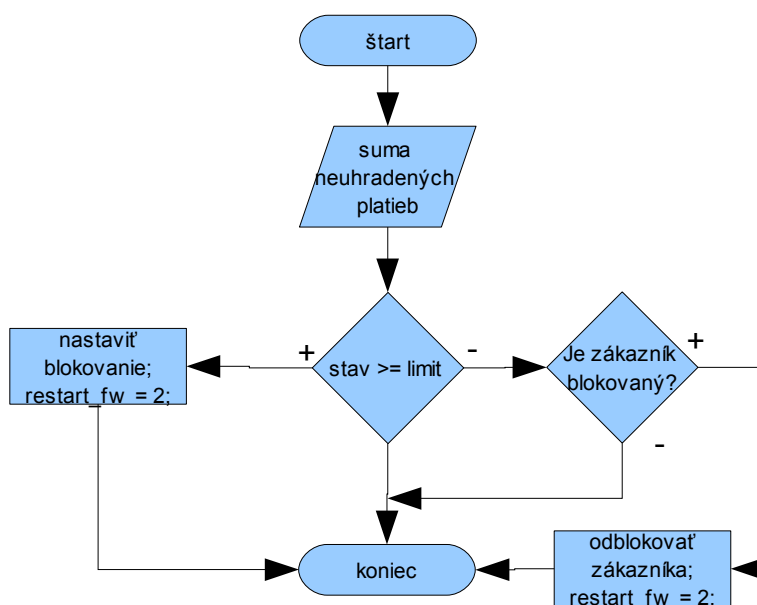
Nevrátenie zmluvy

Je typ blokovania, ktorý informuje zákazníka o nevrátení platnej zmluvy za poskytovanie pripojenia do siete internet. Systém po zaregistrovaní zákazníka automaticky vygeneruje zmluvu na základe vložených informácií. Kompetentný užívateľ spracuje zmluvu prekontrolovaním údajov a odošle zákazníkovi na podpis pričom v systéme označí dátum odoslania. Skript pre kontrolu nevrátených zmlúv sa spúšťa raz za deň a zisťuje či odoslané zmluvy boli vrátené. V prípade že by bol rozdiel medzi dátumami (dátum odoslania a aktuálny deň) väčší ako 14 dní, systém automaticky obmedzí zákazníkovi služby www a informuje ho zobrazením textu s dôvodom blokovania. Toto blokovanie sa automaticky deaktivuje zadáním dátumu doručenia, resp. je ho možné ručne vypnúť.

Skript kontrolujúci zmluvy je v prílohe č. 4.

Neplatič

Je to druhý typ blokovania, ktorý informuje zákazníka o neuhradení pohľadávok splatných za obdobie resp. obdobia. Systém každý mesiac generuje platby k úhrade (viď kap. 4.1.3.1) Pri kontrole systém porovnáva počet neuhradených platieb s limitom (počet platieb ktorý sa ešte toleruje je daný konštantou v skripte). Pri prekročení tohto limitu, skript ktorý prevádza túto kontrolu automaticky obmedzí zákazníkovi služby www ako v predošlom prípade. Tento skript kontroluje stav úhrad každú hodinu, aby pri pripísaní platby na účet zákazníka bola prevádzka internetu ihneď obnovená. Vývojový diagram kontrolného skriptu pre platby je zobrazený na obrázku č. 8.



Obrázok 8: Vývojový diagram kontrolného skriptu

4.1.3 Účtovanie poplatkov

Neoddeliteľnou súčasťou systému je účtovanie a fakturácia poplatkov zákazníkom za poskytované služby. Podľa požiadaviek zadávateľa projektu, systém generuje platby na základe typu zákazníka. Fyzickým osobám na základe uzatvorenia zmluvy medzi ISP a koncovým zákazníkom nie je nutné zasielať faktúry, platia podľa zmluvy. Naopak pre zákazníkov ktorý vykonávajú ekonomickú činnosť systém musí generovať faktúry a posilať na email zákazníka. Tento proces je z celá automatizovaný a nevyžaduje žiadnu interakciu z užívateľa predstavovaného systému.

4.1.3.1 Generovanie platieb

Generovanie platieb na úhradu resp. faktúr prebieha v prvom dni účtovného obdobia, čo je mesiac. Systém automaticky na základe nastavenej tarify alebo v prípade že zákazník využíva dodatkové služby ako je napr. verejná IP adresa, spočíta výslednú sumu a vloží nový záznam (platbu na úhradu) do databázy, ktorý obsahuje pre fyzickú osobu atribúty z požiadavky č. 9. Pre osoby vykonávajúce ekonomickú činnosť je záznam obsiahlejší pretože na jeho základe sa generuje faktúra. Systém zobrazuje generované faktúry dynamicky v čase požiadavky a to na základe záznamu o platbe uloženého v databáze. Kvôli tomuto faktu, systém musí kompletne vytvoriť nový záznam spolu s kontaktnými údajmi zákazníka aby sa neprejavili zmeny i na faktúrach v prípade opravy týchto údajov v profile zákazníka.

4.1.3.2 Úhrada platieb

Systém zavádza osobný účet zákazníka na ktorý sa pripisujú kreditové položky v prípade importu z banky alebo ručným zadáním. Z tohto účtu zákazníka sa skript kontrolujúci jeho stav pokúša periodicky uhradiť vygenerované platby. Skript prechádza doposiaľ neuhradené platby a v prípade dostatočného kreditu potrebného na úhradu platby je platba uhradená a odpísaná z účtu zákazníka.

4.1.3.3 Import z banky

Import platieb z banky je nedeliteľnou súčasťou hocakého fakturačného systému. V tomto projekte bol pevne zadefinovaný formát výstupu z výpisu z banky, kde pohyby na účte majú jednotlivé položky oddelené čiarkou a ich význam je nasledovný

1. dátum pohybu na účte (deň.mesiac.rok)
2. číslo účtu ku ktorému sa vzťahuje výpis (číslice)
3. správa pre prijímateľa (reťazec v úvodzovkách)
4. číslo účtu platiteľa (číslice)
5. kód banky platiteľa (číslice)
6. suma obratu (reťazec v úvodzovkách)
7. mena obratu (reťazec)
8. variabilný symbol (číslice)
9. konštantný symbol (číslice)
10. špecifický symbol (číslice)

Vzor pohybu na výpise z účtu:

27.04.09,2469572259,"april",6244849112,0200,"+11,62",EUR,151,,,

Na základe hore uvedeného vzoru bol vytvorený nasledovný regulárny výraz, ktorý je postupne aplikovaný na všetky riadky výpisu:

```
^([0-9]{2})\.([0-9]{2})\.([0-9]{2}),.*, "(.*)", ([0-9]*-[0-9]+), ([0-9]{4}), "(.*)", ([A-Z]*), ([0-9]*), ([0-9]*), ([0-9]*),.*$
```

kde nás zaujímajú všetky položky okrem položky č. 2. Vhodným parsovaním takéhoto importu sa platby pripíšu na osobný účet zákazníka v prípade zhody VS s ID zákazníka v databáze. Ak by tak nedošlo takáto platba sa označí ako nespárovaná, kde sa ďalej očakáva akcia zo strany užívateľa systému (ručné dohľadanie a pod.) Systém taktiež rieši problém s duplicitnými platbami ktoré by mohli nastať či už zo strany platiteľa alebo užívateľa v importovaní 2x toho samého súboru. Preto sú všetky pohyby ukladané do databázy spolu s hash hodnotou a pred importovaním sa kontroluje či už v systéme nie je rovnaká platba. V prípade zhody nie je takáto platba automaticky pripísaná na účet zákazníka ale je označená ako nespárovaná a odložená na prešetrenie. Samotná logika importu je vložená do modulu pre import vo web rozhraní systému. (viď programová dokumentácia na priloženom CD)

4.1.4 Štatistiky a grafy

4.1.4.1 Zdroje dát

Pre zber štatistík bol zvolený voľne dostupný nástroj IPFM⁸ (IP Flow Meter), ktorý merá šírku pásma klientov pripojených do siete internet. Nástroj má možnosť nastavenia periódy po ktorej bude ukladať namerané dáta do súboru. Na základe požiadavky č. 33 bol zvolený 5 minútový cyklus.

Vzor výstupného súboru:

HOST	IN	OUT	TOTAL
192.168.1.1	12345	6666684	6679029
192.168.1.2	1232314	12345	1244659
192.168.1.3	6645632	123	6645755

4.1.4.2 Záznam údajov

Vo výstupnom súbore jednotlivé riadky obsahujú IP adresu klienta, prenesené dáta download, upload a total v bajtoch. Po výpise do súboru sa počítadlá vynulujú a záznamy uložia do databázy. Do databázy sa ukladá namiesto IP adresy identifikácia prípojky zákazníka. To je nevyhnutné z toho hľadiska, pokiaľ by sa zmenila IP adresa klienta uložené štatistiky by neboli prístupné a takýmto spôsobom zostávajú naďalej konzistentné.

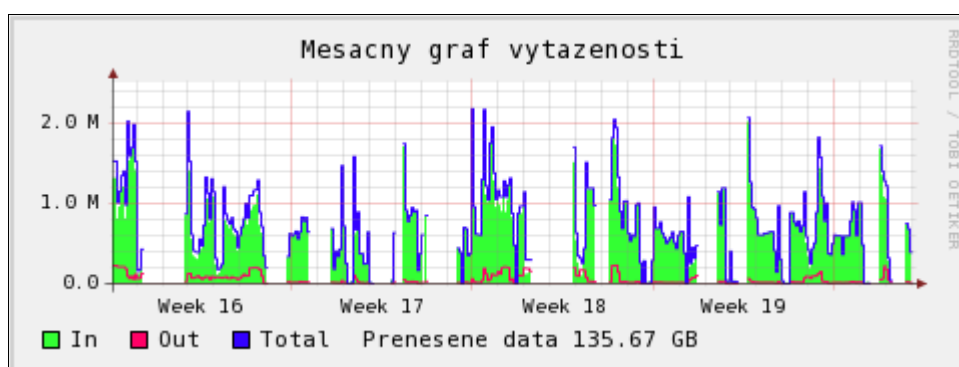
Aby sa predišlo pri veľkom počte klientov zaplnením dostupných zdrojov, databáza pre štatistiky obsahuje 2 tabuľky. Do jednej sa ukladajú dáta s požadovanou periódou a slúži k zobrazeniu štatistík prenesených dát za aktuálny deň. Po polnoci sa všetky namerané údaje pre klienta spočítajú, vypočíta sa priemerná rýchlosť a zapisujú ako nový záznam do ďalšej tabuľky, ktorá reprezentuje

8 <http://robert.cheramy.net/ipfm/>

jednotlivé merané dni. Skript parsujúci daný výstup súboru a vykonávajúci záznam údajov je v prílohe č. 5.

4.1.4.3 Grafy s pomocou RRDtool

Zobrazenie priebehu zaznamenaných hodnôt v čase tvorí základ pre jednoduchú a užívateľsky prehľadnú štatistiku prenesených dát zákazníka. Systém používa pre záznam a zobrazenie rôznych pohľadov na štatistiky (napr. Mesačný graf vyťaženia vid' obr. 9) nástroj RRDtool [9]. RRDtool je založený na princípe databázy statickej veľkosti kde v tzv. Round Robin archívoch sú definované dátové zdroje, ktorých údaje sa zaznamenávajú v predom daných rozlíšeniach v rámci stanovených časových rozsahov. Systém pre každú prípojku zákazníka vytvára štyri RRD databázy, ktoré sú v časových horizontoch jedného dňa, týždňa, mesiaca a roka. Vytvorenie týchto databáz a ich aktualizácia je taktiež obsiahnutá v hore spomínanom skripte.



Obrázok 9: Ukážka grafu pre prípojku zákazníka

4.2 Implementácia a použité technológie

4.2.1 Programovací jazyk

S ohľadom na požiadavku č. 37, aby bol systém dostupný z internetového prehliadača, ponúka sa široká paleta technológií a jazykov. Aby sme však vyhovelí p.č. 38-39 bol zvolený ako programovací jazyk PHP⁹, ktorý je ľahko dostupný a platformovo nezávislý. PHP je skriptovací jazyk a v súčasnosti veľmi rozšírená technológia umožňujúca programovanie na strane serveru s veľkým počtom rozširujúcich modulov. Pre prezentáciu dokumentu v značkovacom jazyku XHTML je v tejto práci využitý šablónovací systém Smarty¹⁰, ktorý umožňuje oddeliť aplikačnú logiku od prezentácie dát, čo zaisťuje prehľadnejší a ľahko editovateľný zdrojový kód. Výstupnú formu dokumentom dodávajú štýly CSS. Pre prirodzenejšie užívateľské rozhranie bola i tu využitá technológia JavaScript, ktorá umožňuje zmenu dokumentu na strane klienta. Dynamické generovanie PDF súborov je zabezpečené

9 <http://www.php.net/>

10 <http://www.smarty.net/>

externou knižnicou FPDF¹¹ pre zvolený programovací jazyk. O automatizované operácie a periodické spúšťanie skriptov sa stará démon Cron spustený na strane smerovača. Príklad s konfiguráciou Cron démona je súčasťou prílohy č. 6.

Nakoľko je predstavovaný systém komplexnejšieho charakteru, bola vytvorená programová dokumentácia ktorá obsahuje popis modulov systému a je obsahom priloženého CD spolu s užívateľským manuálom a postupom inštalácie.

4.2.2 Databázový systém

Pre účely ukladania a správy dát bol zvolený databázový systém MySQL. Je to multiplatformová databáza optimalizovaná na rýchlosť (úložisko typu MyISAM) a predovšetkým bezproblémová komunikácia s mnohými programovacími jazykmi. S rastúcimi požiadavkami na funkcionality sa i tu vyvinuli vyššie funkcie ako je napríklad ukladanie procedúr ktoré sú i v tomto projekte využité. Práca s databázou je v tomto projekte veľmi kľúčová a pre uľahčenie práce s ňou bola vytvorená trieda, ktorá obsahuje sadu funkcií operujúc nad zvolenou databázou a umožňuje tak transparentný prístup k údajom.

Štruktúra databázy je zobrazená v dátovom diagrame v prílohe č. 3.

4.2.3 Štruktúra projektu

Predstavovaný systém je tvorený a implementovaný sadou PHP skriptov, ktoré si prostredníctvom databázy vymieňajú informácie a tak tvoria stabilný systém ktorý spĺňa všetky kladené požiadavky. Celý projekt je rozdelený, ako už bolo spomenuté v predchádzajúcich kapitolách, na dve časti, ktoré sú nasadené na dva rozdielne stroje a ich hierarchia adresárov je zhrnutá v nasledujúcich podkapitolách.

4.2.3.1 Web rozhranie

- / – koreňový adresár obsahuje súbor index.php pre spustenie rozhrania a konfiguračný súbor
- /includes – adresár obsahujúci súbory s triedami a pomocnými funkciami
- /images – adresár s obrázkami pre web rozhranie
- /js – adresár obsahujúci externé súbory s JavaScript funkciami
- /css – adresár so súbormi CSS štýlov
- /modules – adresár so súbormi s aplikačnou logikou pre jednotlivé moduly systému
- /templates – adresár s HTML šablónami pre vykreslenie jednotlivých modulov
- /templates_c – adresár potrebný pre skompilované šablóny

4.2.3.2 Skripty pre router

- /includes – adresár s triedou pre databázu a konfiguráciami
- /redirect – obsluha blokovania zákazníka – dynamicky generované stránky
- /system – jadro systému, adresár obsahuje sadu skript pre automatizované akcie
- /traffic – adresár so skriptami pre získavanie a aktualizáciu štatistík

¹¹ <http://www.fpdf.org/>

5 Záver

Cieľom tejto práce bolo navrhnúť a implementovať systém pre správu komerčnej siete. Nutnosť nasadenia takéhoto systému pre poskytovateľa internetových služieb je naznačený úvodom tejto práce. Návrh komplexného systému informačného charakteru vychádzal z nedostatkov súčasného riešenia skutočnej počítačovej siete, ktorá má ako hlavnú bránu do siete internet server s OS Linux.

Použité technológie a nástroje pre vytvorenie brány, cez ktorú sú pripojení klienti na internet, boli rozobrané v druhej kapitole. Táto kapitola sa taktiež zaoberala zabezpečením šírky pásma, kde bol nasadený voľne dostupný program pod GPL licenciou. Nakoniec boli zhrnuté nedostatky tohto riešenia, hlavne čo sa týka správy a evidencie klientov.

Na základe detailnejšej analýzy požiadaviek z tretej kapitoly bol navrhnutý systém a jeho grafické rozhranie zobraziteľného v internetovom prehliadači. V tejto kapitole bola popísaná najmä požadovaná funkčnosť a vybrané prípady použitia systému.

Štvrtá kapitola uvádza riešenie a postupy realizácie kľúčových častí navrhnutého systému a odkazuje už na konkrétne implementácie popísaných postupov. Pre ilustráciu grafického rozhrania boli do textu začlenené i obrázky z vyvíjaného IS. Kapitola uzatvára časť popisujúca použité programovacie prostriedky, ktoré boli zvolené pre implementáciu systému pre správu siete.

Vyvíjaný systém bol nasadený do reálnej prevádzky ihneď po vytvorení jeho prototypu, čo sa veľmi osvedčilo pri odhaľovaní jeho menších nedostatkov s možnosťou rýchlej nápravy. Takto boli postupne splnené všetky požiadavky kladené na tento systém a jeho finálna verzia zabezpečuje v súčasnosti bezproblémový chod a správu siete s takmer tisíc pripojených klientov.

5.1 Možnosti ďalšieho rozšírenia

Nakoľko bol predstavený systém vyvinutý pre konkrétne požiadavky, jeho súčasná verzia neumožňuje rôzne typy nastavení (napr. počet záznamov na stránku pri listovaní zoznamom, definovanie užívateľských práv pre rôzne moduly systému a i.) z grafického rozhrania. Do budúcnosti sa predpokladá, okrem doplnení spomínaných vlastností i s rozvojom systému v podobe ďalších modulov ktoré by implementovali nasledujúce funkcie:

- rozhranie pre zasielanie hromadnej pošty na základe vybraných filtrov
- vytvoriť a integrovať zákaznícku sekciu spolu s helpdeskom do vyvinutého systému
- aktívna komunikácia s aktívnymi prvkami siete, za účelom aplikovania napr. port-security na switchoch
- monitorovanie a dohľad nad aktívnymi prvkami siete

V rámci uvedených rozšírení je možné predpokladať i s novou implementáciou smerovača, ktorá by zaistila zvýšenie výkonu s narastajúcim sa počtom pripojených klientov.

Literatúra

- [1] Welsh, M., Dalheimer, M.K..Používáme Linux. Computer Press, Brno, 2003.
ISBN 80-7226-698-5
- [2] Král, J. Informační systémy, Science, Veletiny, 1998. ISBN 80-86083-00-4.
- [3] Welling, L., Thomson, L. PHP a MySQL rozvoj webových aplikací. Softpress, Praha, 2002.
ISBN 8086497836
- [4] Sklar, D. PHP - moduly, rozšíření a akcelerátory. Zoner Press, Brno ISBN 80-86815-19-6
- [5] Dokomuntácia šablónovacieho systému Smarty. [online], 2009, [cit. 2009-03-20].
Dostupné z URL: <<http://www.smarty.net/>>
- [6] Hubert, B. Linux Advanced Routing & Traffic Control HOWTO. [online], 2004, [cit. 2009-03-20].
Dostupné z URL: <<http://lartc.org/howto/>>
- [7] Kolektív netfilter: Dokomuntácia firewallu iptables. [online], [cit. 2009-03-20].
Dostupné z URL: <<http://www.netfilter.org/documentation/>>
- [8] Devera, M. HTB Linux queuing discipline manual - user guide. [online], 2002, [cit. 2009-03-20].
Dostupné z URL: <<http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm>>
- [9] Oetiker, T. RRDtool Documentation. [online], 2009, [cit. 2009-05-19].
Dostupné z URL: <<http://oss.oetiker.ch/rrdtool/doc/index.en.html>>
- [10] Pavel Míča: Monitorovací a dohledový systém počítačové sítě, diplomová práce, Brno, FIT VUT v Brně, 2007

Zoznam príloh

Príloha 1. Proces spracovania paketov firewalldom iptables

Príloha 2. Use case model

Príloha 3. Dátový diagram návrhu databázy

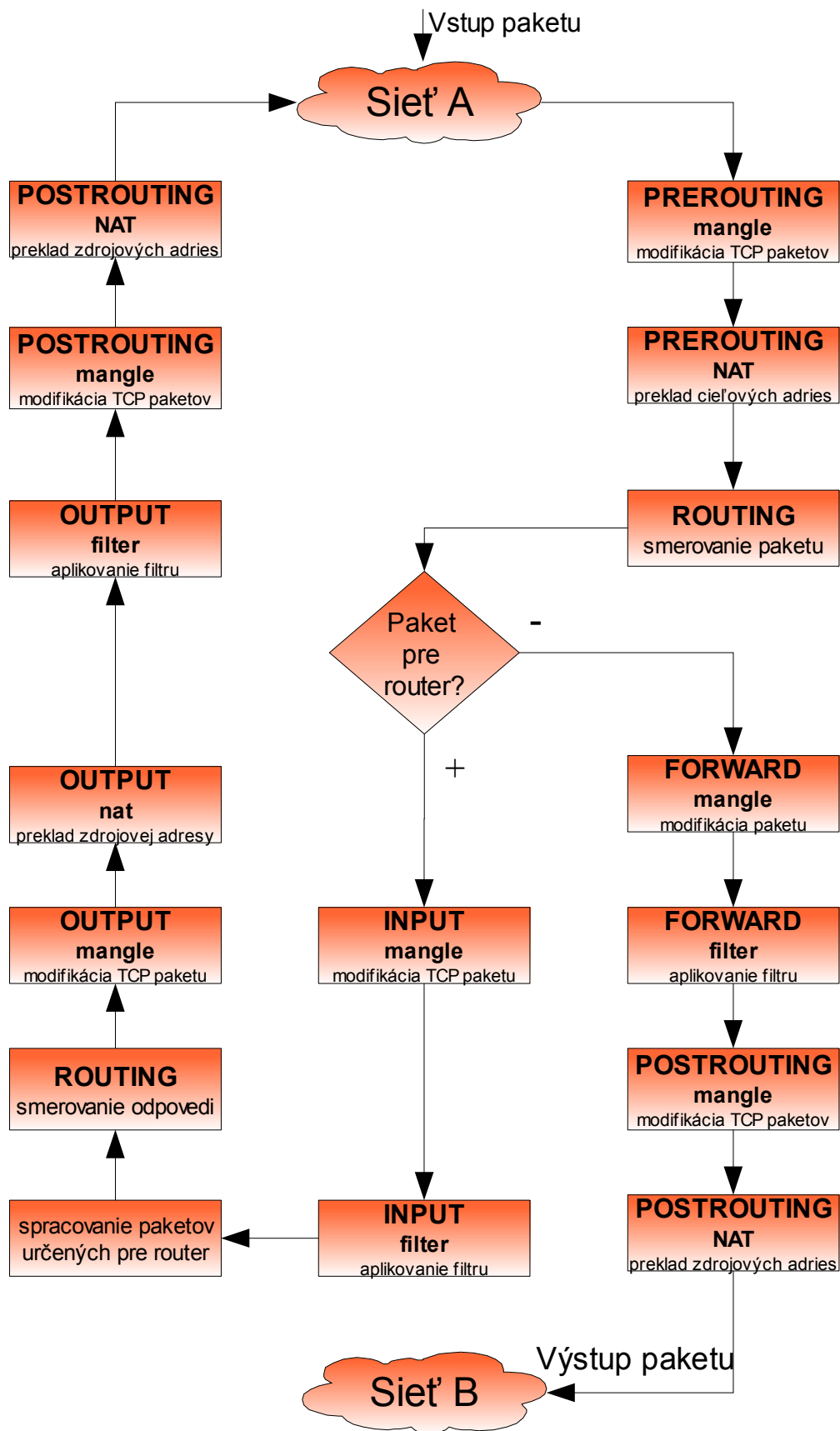
Príloha 4. PHP skript pre automatické blokovanie – nevrátená zmluva

Príloha 5. PHP skript pre parsovanie a záznam štatistík

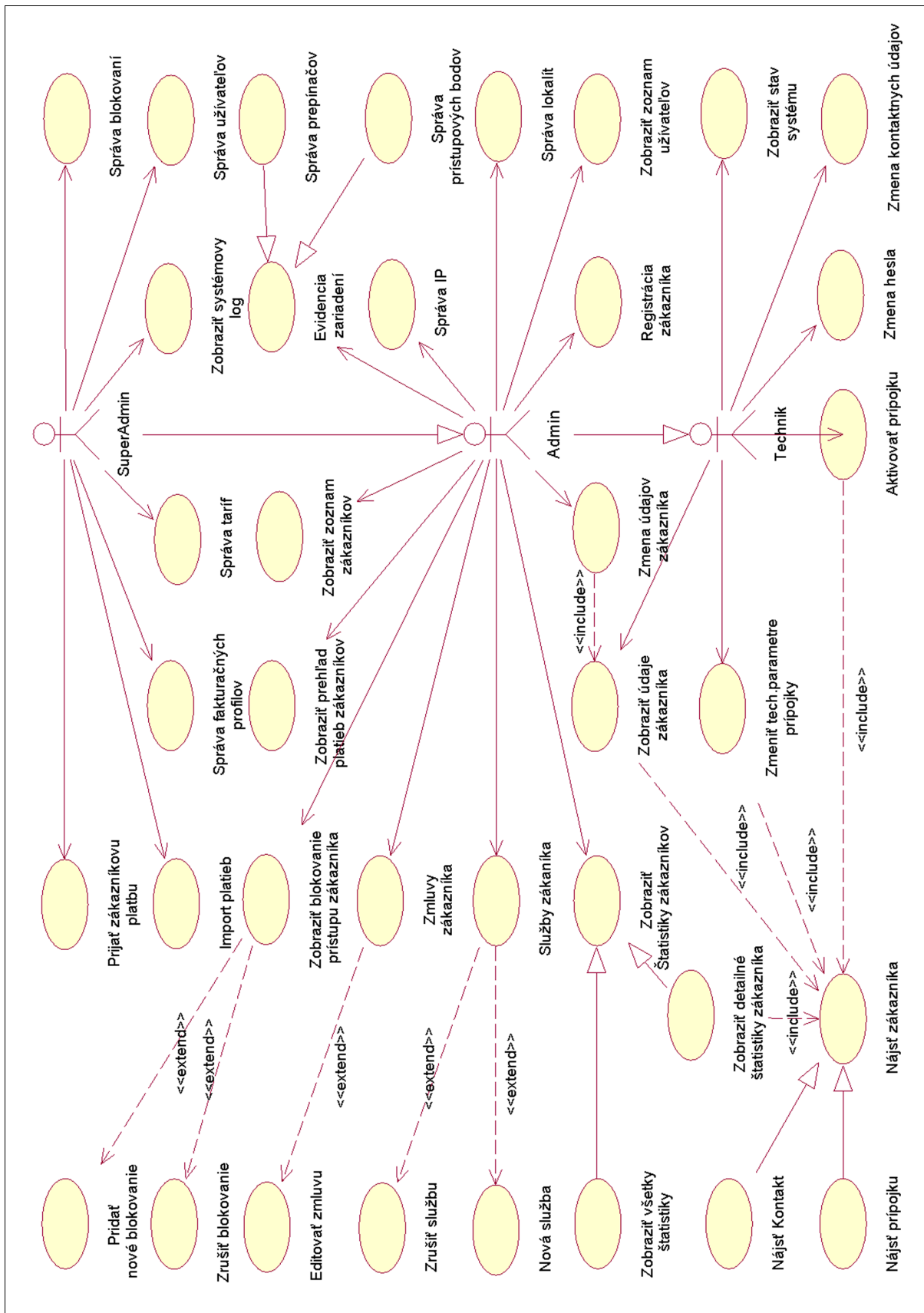
Príloha 6. Príklad konfigurácie cron démona

Príloha 7. CD so zdrojovými kódmi, programovou dokumentáciou a užívateľskou príručkou

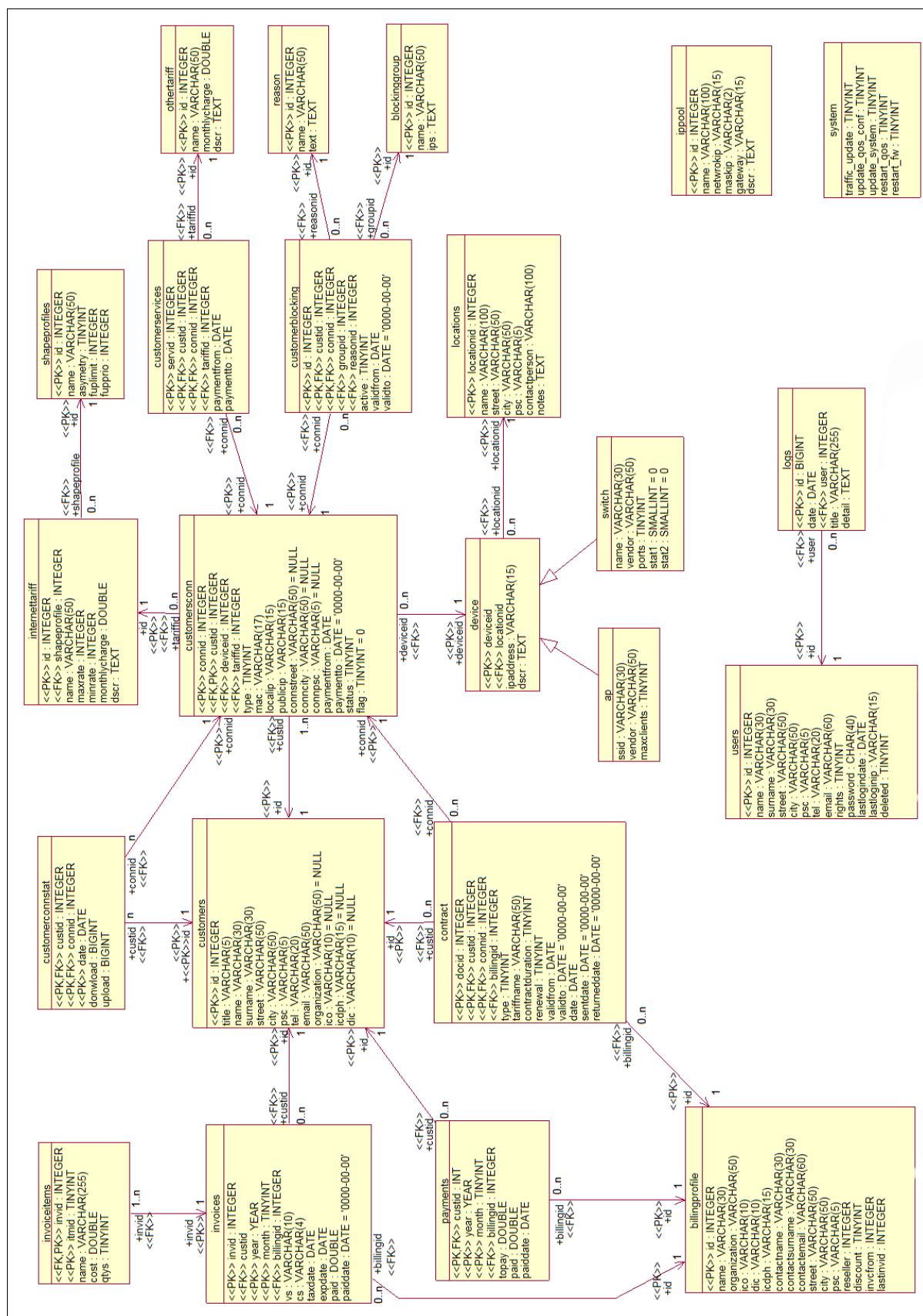
Príloha 1. Proces spracovania paketov firewallom iptables



Príloha 2. Use case model



Príloha 3. Dátový diagram návrhu databázy



Príloha 4. PHP skript pre automatické blokovanie – nevrátená zmluva

```
require("config.php"); // nastavenia a instancia databazy
$DAYS = 14; // nastavenie poctu dni do ktoreho maju byt vratene zmluvy

function blockConn($custid,$connid)
{ // funkcia nastavi blokovanie pre pripojku pre vsetky adresy s dovodom nevratena_zmluva
    global $DB;
    $DB->execute('UPDATE customersconn SET flag=1 WHERE customerid=? AND
connid=?',array($custid,$connid));
    $DB->execute('INSERT INTO customersblocking
(`custid`,`connid`,`group`,`ips`,`reason`,`activatedby`,`activateddate`,`active`,`validfrom`)
VALUES(?,?,0,\'0/0/0\',1,1,NOW(),1,NOW())',array($custid,$connid));
    // nastavenie priznaku pre restart fw
    $DB->execute('UPDATE system SET restart_fw=2');
}

function unBlockConn($custid,$connid)
{ // odblokuje pripojku
    global $DB;
    $DB->execute('UPDATE customersconn SET flag=0 WHERE customerid=? AND
connid=?',array($custid,$connid));
    $DB->execute('UPDATE customersblocking SET active=0,validto=NOW() WHERE reason=1 AND
custid=? AND connid=? AND ips=\'0/0/0\'',array($custid,$connid));
    $DB->execute('UPDATE system SET restart_fw=2');
}

// zoznam pripojok pre ktore neboli doposial vratene zmluvy od dna odoslania
$notreturned = $DB->getAll('SELECT custid,connid FROM contracts WHERE type=1 AND (sentdate
IS NOT NULL OR sentdate!=0) AND DATEDIFF(DATE(NOW()),DATE(sentdate))>? AND (returneddate IS
NULL OR returneddate=0)',array($DAYS));
if(count($notreturned))
foreach($notreturned as $client)
{ // ak je pripojka uz blokovana -> neblokovať
    if(!$DB->getOne('SELECT flag FROM customersconn WHERE customerid=? AND
connid=?',array($client['custid'],$client['connid'])))
        blockConn($client['custid'],$client['connid']);
}

// zoznam blokovanych pripojok ktore vratili zmluvu -> odblokovať
$blocked = $DB->getAll('SELECT CONN.customerid,CONN.connid FROM customersconn CONN WHERE
CONN.flag=1 AND (SELECT C.returneddate FROM contracts C WHERE C.type=1 AND
C.custid=CONN.customerid AND C.connid=CONN.connid)');
foreach($blocked as $unblock)
{
    unBlockConn($unblock['customerid'],$unblock['connid']);
}

// zatvorenie spojenia s databazou
$DB->disconnect();
```

Príloha 5. PHP skript pre parsovanie a záznam štatistík – ukážka zdrojového kódu

```
... nastavenie ciest ...

/* KONIEC NASTAVENI */
/*****
$DB = new Database($DB_SERV,$DB_USR,$DB_PASS,$DB_DB); // vytvorenie instance databazy
$data = file($inputfile); // vstupny subor z ipfm exportu

foreach ($data as $row)
{ // prejde kazdy riadok suboru
    // parsovanie riadku a ziskanie IP adresy, prenesene data download/upload, total
    preg_match('/(([0-9]+\.){3}[0-9]+\s+([0-9]+\s+([0-9]+\s+([0-9]+\s+([0-9]+\s+.*\/', $row, $matches);
    $ip = $matches[1];
    $input = $matches[3];
    $output = $matches[4];
    $total = $matches[5];
    // ziskanie identifikacie pripojky na zaklade IP adresy zo suboru
    $customer = $DB->getRow('SELECT customerid,connid FROM customersconn WHERE
localip=?',array($ip));
    // ak sa dana IP nenachadza v DB neprebehne zaznam a ide sa parsovat dalsi riadok zo sub.
    if(count($customer)!=2) continue;

    // operacie nad RRD databazmi
    // vytvorenie nazvu suboru pre rrd
    $prefix = $customer['customerid'].'-'.$customer['connid'];
    if(!file_exists($rrd_path."rrds/".$prefix."-day.rrd"))
    { // ak taky subor este neexistuje vytvori sa s pozadovanymi parametrami
        $cmd = "rrdtool create ".$rrd_path."rrds/".$prefix."-day.rrd --start N --step 300
DS:In:ABSOLUTE:7200:0:U DS:Out:ABSOLUTE:7200:0:U DS:Total:ABSOLUTE:7200:0:U
RRA:AVERAGE:0.5:1:288";
        exec($cmd);
    }
    // priprava prikazu pre vloženie vzorky statistik do rrd / update
    $cmd = "rrdtool update ".$rrd_path."rrds/".$prefix."-day.rrd N:".$input.":".$output.":".$total;
    // vykonanie prikazu
    exec($cmd);

    // zapis aktualne prenesenych dat i do databazy mysql
    $DB->execute('INSERT INTO customersconnstatday (custid,connid,time,download,upload)
VALUES(?,?,NOW(),?,?)',
        array($customer['customerid'],$customer['connid'],$input,$output));
}

// ziska rozdiel medzi aktualnym dnom a poslednym prenosom do celkovych statistik
$res=$DB->getOne('SELECT DATEDIFF( DATE(NOW()),DATE(traffic_update)) as delta FROM system');
if($res['delta']>0)
{ // ocitli sme sa v novom dni budeme prenasat do celkovych statistik
    // zratanie up/down a vypocet avg rate
    //custid,connid,date,download,upload,avgrate
    $DB->execute('INSERT INTO customersconnstat (SELECT
custid,connid,SUBDATE( DATE(NOW()),1),SUM(download),SUM(upload),AVG(download)/5/60*8,AVG(uploa
d)/5/60*8 FROM customersconnstatday GROUP BY custid,connid)');
    // vyprazdnenie novej tabulky
    $DB->execute('TRUNCATE TABLE customersconnstatday');
    // aktualizacia dna prenosu
    $DB->execute('UPDATE system SET traffic_update=NOW()');
}
// uzatvorenie spojenia s DB
$DB->disconnect();
```


Príloha 6. Príklad konfigurácie cron démona na strane smerovača

```
#kontrola priznakov systemu kazdu minutu
#nastavovanie pravidiel firewallu, generovanie konfiguracnych suborov pre QoS
* * * * * php /var/www/system/systemcheck.php

#parovanie a aktualizacia statistik kazdych 5 minut
*/5 * * * * php /var/www/traffic_counter/update.php

#generovanie platieb a rozosielanie emailov s fakturami prveho dna mesiaca 10 min. po polnoci
10 0 1 * * php /var/www/system/genpayments.php

#skript provadi kontrolu nevratenych zmluv kazdu hodinu o 5 minute
5 * * * * php /var/www/system/checkcontracts.php

#sparovanie platby ak ma zakaznik na ucte dostatok prostriedkov - kontrola kazdu hodinu
5 * * * * php /var/www/system/checkaccounts.php

#skript provadi kontrolu stavu pohladavok zakaznika kazdu hodinu o 6 minute
6 * * * * php /var/www/system/checkpayments.php

#automaticke odpojenie pripojky zakaznika po zadani datumu - kontrola kazdy den o 1 hodine
0 1 * * * php /var/www/system/expiredusers.php
```